

TUGAS AKHIR - TM 141585

PEMODELAN DAN RANCANG BANGUN SISTEM PENGENDALIAN PID DAN *SLIDING MODE* *CONTROL* UNTUK *REMOTE CONTROL* *WEAPON STATION* (RCWS)

HANS KRISTIAN
NRP 2113 100 005

Dosen Pembimbing
Ir. Bambang Pramujati, MSc.Eng, Ph.D.
Hendro Nurhadi, Dipl. Ing., Phd.

Departemen Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR – TM141585

**PEMODELAN DAN RANCANG BANGUN SISTEM
PENGENDALIAN PID DAN *SLIDING MODE CONTROL*
UNTUK *REMOTE CONTROL WEAPON STATION*
(RCWS)**

**HANS KRISTIAN
NRP. 2113 100 005**

PEMBIMBING:

**Ir. Bambang Pramujati, MSc.Eng, PhD.
Hendro Nurhadi, Dipl. Ing., Ph.D.**

PROGRAM SARJANA
JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017



FINAL PROJECT – TM141585

**MODELLING AND DESIGNING CONTROLLER
SYSTEM USING PID AND SLIDING MODE CONTROL
FOR REMOTE CONTROL WEAPON STATION (RCWS)**

**HANS KRISTIAN
NRP. 2113 100 005**

SUPERVISORS:

**Ir. Bambang Pramujati, MSc.Eng, PhD.
Hendro Nurhadi, Dipl. Ing., Ph.D.**

BACHELOR DEGREE PROGRAM SARJANA
MECHANICAL ENGINEERING DEPARTMENT
FACULTY OF INDUSTRIAL
SEPULUH NOPEMBER INSTITUTE OD TECHNOLOGY
SURABAYA 2017

**PEMODELAN DAN RANCANG BANGUN SISTEM
PENGENDALIAN PID DAN *SLIDING MODE CONTROL*
UNTUK REMOTE CONTROL WEAPON STATION
(RCWS)**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada





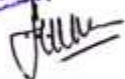
Program Studi S-1 Departemen Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

HANS KRISTIAN

NRP. 2113 100 005

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Bambang Pramujati, MSc.Eng.  (Pembimbing I)
Ph.D.
NIP. 196912031994031001
2. Hendro Nurhadi, Dipl. Ing., Ph.D.  (Pembimbing II)
NIP. 197511202002121002
3. Unggul Wasiwitono, ST.,  (Penguji I)
M.Eng.SC.,Dr.Eng.
NIP. 197805102001121001
4. Latifah Nurahmi, ST., M.Sc., Ph.D.  (Penguji II)
5. Ari Kurniawan Saputra S.T., M.T.  (Penguji III)
NIP. 198604012015041001

SURABAYA

JULI, 2017

**PEMODELAN DAN RANCANG BANGUN SISTEM
PENGENDALIAN PID DAN *SLIDING MODE CONTROL*
UNTUK *REMOTE CONTROL WEAPON STATION*
(RCWS)**

Nama : Hans Kristian

NRP : 2113 100 005

**Pembimbing : Ir. Bambang Pramujati, MSc.Eng, PhD.
Hendro Nurhadi, Dipl. Ing., Ph.D.**

Abstrak

Perang merupakan pertempuran bersenjata antara dua pasukan atau lebih (KBBI). Peperangan dapat terjadi di darat, laut dan udara tetapi wilayah darat merupakan tujuan perebutan karena merupakan pusat pemerintahan. Untuk menghadapi peperangan darat maka kendaraan tempur dengan teknologi baru merupakan ALUTSISTA yang wajib ada. Berdasarkan UU no 3 tahun 2002 tentang pertahanan Negara menyebutkan bahwa Negara memiliki kewajiban mempersiapkan dan mengembangkan ALUTSISTA untuk berjaga-jaga jika terjadi ancaman Negara. Pengembangan akan dilakukan pada penelitian ini yaitu pada pemodelan sistem dan pengendalian *Remote Control Weapon Station* (RCWS).

Penelitian ini merupakan salah satu sub tim yang bekerja sama dalam pengembangan RCWS. Tugas dari penelitian ini adalah untuk memodelkan dan membangun pengendalian yang sesuai untuk RCWS 12,7mm. RCWS 12,7mm dapat dianggap sebagai sebuah manipulator seri 2 *revolute joint* dengan pergerakan ruang. Metode *inverse kinematic* digunakan untuk menghitung permodelan kinematis. Hasil dari *inverse kinematic* adalah besarnya sudut geser yang diperlukan dari masing-masing sendi dan transformasi matriks yang berfungsi sebagai gambaran matematis pergerakan antar lokal koordinat. Selanjutnya, model kinematis dan matriks transformasi diolah menjadi model dinamik dengan formulasi lagrangian. Permodelan dinamis diolah

menjadi *state space* agar dapat disimulasikan pada perangkat lunak Simulink bersama dengan pengendali.

Pengendalian pada RCWS menggunakan metode PID dan *sliding mode control* (SMC). Penentuan parameter PID menggunakan optimasi GRG Nonlinear sedangkan penentuan parameter untuk SMC menggunakan hasil dari simulasi. SMC memiliki keunggulan dalam mengendalikan sistem dinamik nonlinear yang kompleks. Dari hasil simulasi dan perbandingan maka dipilih pengendali SMC dengan keunggulan pada semua parameter kontrol. Respon transien RCWS 12.7 yaitu nilai *settling time* sebesar 0.875 sekon, % *Overshoot* sebesar 3.65% dan *RMS Error* sebesar 0.003 rad. Pengendali SMC juga terbukti *robust* walaupun telah ditambahkan gangguan eksternal dengan *settling time* setelah penambahan gangguan kurang dari 0.5 sekon.

Kata kunci: RCWS, *Serial Manipulator*, *Inverse Kinematic*, *Lagrangian Formula*, *PID controller*, *Sliding Mode Control*, *Robust*, *Disturbance*.

MODELLING AND DESIGNING CONTROLLER SYSTEM USING PID AND SLIDING MODE CONTROL FOR REMOTE CONTROL WEAPON STATION (RCWS)

Name : Hans Kristian
NRP : 2113 100 005
Supervisors : Ir. Bambang Pramujati, MSc.Eng, PhD.
Hendro Nurhadi, Dipl. Ing., Ph.D.

Abstract

War is an armed battle between two or more troops (KBBI). Warfare can occur on land, sea and air but the land territory is a central point of struggling because it is where the center of the government is running. To face a warfare on the land, a new combat vehicle with the current technology is ALUTSISTA that need to exist. The Law No. 3 of 2002 on State Defense states that the State has the obligation to prepare and develop ALUTSISTA in case of a State threat. Development will be done in this research, more specifically on the modeling system and the control of Remote Control Weapon Station (RCWS).

This research is one of the sub-teams that are working together in the development of RCWS. The task of this research is to model and design the appropriate controller for 12.7mm RCWS. 12.7mm RCWS can be regarded as a revolute joint 2 series manipulator with space movement. The inverse kinematic method is used to calculate kinematic modeling. The result of the inverse kinematic is the magnitude of the required shear angle of each joint and the matrix transformation which serves as a mathematical representation of the movements between local coordinates. Furthermore, the kinematic model and transformation matrix are processed into dynamic models with lagrangian formulations. Dynamic modeling is processed into state space in order to simulate the simulink software together with the controller.

Control on RCWS uses PID and sliding mode control (SMC) methods. The determination of PID parameter used the Nonlinear GRG optimization while the determination of SMC parameter used the result from the simulation. SMC has an advantage in controlling the complex nonlinear dynamic systems. From the simulation and the comparison of the results, then SMC controller was selected with advantages for all control parameters. The transient response of RCWS 12.7, which is the settling time value of 0.875 sekon, % Overshoot of 3.65% and RMS Error of 0.003 rad. SMC controllers were also proven to be robust despite the added external disturbance with settling time after the addition of disturbance less than 0.5 seconds.

Keywords: RCWS, Serial Manipulator, Inverse Kinematic, Lagrangian Formula, PID controller, Sliding Mode Control, Robust, Disturbance.

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa karena atas penyertaanya penulis dapat menyelesaikan Tugas Akhir dengan judul **“Pemodelan dan Rancang Bangun Sistem Pengendalian PID dan *Sliding Mode Control* untuk *Remote Control Weapon Station* (RCWS)”**.

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan dan memperoleh gelar sarjana pada Jurusan Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS). Dalam menyelesaikan Tugas Akhir ini penulis banyak mendapat bantuan dan masukan dari berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Kedua orangtua dan koko yang telah memberikan semangat dan doa yang berlimpah untuk menyelesaikan Tugas Akhir ini
2. Bapak Ir. Bambang Pramujati, MSc.Eng, PhD. selaku ketua jurusan Teknik Mesin FTI ITS, dosen wali penulis dan dosen pembimbing I atas dukungan nasihat dan bimbingannya selama masa kuliah maupun dalam penulisan Tugas Akhir ini.
3. Bapak Hendro Nurhadi, Dipl. Ing., Ph.D., selaku dosen pembimbing II yang juga setia memberikan bimbingan dan masukan untuk menyelesaikan Tugas Akhir.
4. Bapak Unggul Wasiwitono, ST., M.Eng.SC.Dr.Eng. dan ibu Latifah Nurahmi, ST., M.Sc., Ph.D. sebagai dosen Teknik Mesin FTI ITS yang juga memberikan masukan dan arahan dalam penyelesaian Tugas Akhir ini.

5. Seluruh dosen pengajar dan karyawan Jurusan Teknik Mesin yang telah banyak memberikan ilmu dan membantu penulis selama menempuh perkuliahan.
6. Bill Febrian W dan Jeffrey sebagai teman 1 lab otomasi yang senantiasa membantu dan berbagi cerita dalam pengerjaan topik otomasi yang jarang sekali mahasiswa Teknik Mesin mengambil topik ini.
7. Semua teman-teman dekat saat SMA maupun kuliah yang senantiasa memberikan dukungan moril saat pengerjaan Tugas Akhir ini.
8. Semua teman-teman Mesin angkatan 2013 atas pengalaman yang berkesan
9. Dan semua pihak yang tidak dapat disebutkan satu persatu.

Harapan penulis semoga Tugas Akhir ini bermanfaat bagi para pembaca. Penulis menyadari bahwa tugas akhir ini jauh dari sempurna. Oleh karena itu, penulis sangat terbuka untuk saran dan kritik dari semua pihak untuk tahapan pengembangan selanjutnya yang lebih baik.

Surabaya, 27 Juli 2017
Penulis

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II.....	5
2.1 Penelitian Terdahulu	5
2.2 Permodelan Kinematis	6
2.2.1 Transformasi pada manipulator robot	6
2.2.2 Analisa posisi pada Manipulator Seri.....	10
2.2.3 Parameter Denavit-Hartenberg (<i>DH Parameter</i>)	11
2.2.4 Direct Kinematic.....	12
2.2.5 Inverse Kinematic	13
2.2.6 Jacobian Matriks.....	13
2.3 Permodelan Dinamis	15
2.3.1 <i>Link Inertia Matrices</i>	15

2.3.2 Manipulator Inertia Matrices (M)	16
2.3.3 Velocities Coupling Vector (V_i)	16
2.3.4 Gravitational Vector (G_i)	16
2.3.5 Formula Lagrangian.....	17
2.4 Pemodelan <i>State Space</i>	17
2.5 Pengendalian	20
2.5.1 Time Response	20
2.5.2 Desain Kontroler PID	21
2.5.3 Rancangan Percobaan <i>Response Surface</i>	23
2.5.3 Analisis Regresi	24
2.5.4 Optimasi GRG Non-linier	25
2.5.5 <i>Sliding Mode Control</i>	25
BAB III	29
3.1 <i>Flowchart</i> Penelitian	29
3.2 Metode Penelitian.....	30
3.2 Parameter RCWS 12.7 mm.....	31
3.3.1 Dimensi dan Desain RCWS 12.7 mm.....	31
3.3.2 Desain Motor dan <i>Gearbox</i>	33
3.3.3 Desain Kriteria Kontrol Alat	33
3.4 <i>Flowchart</i> Pemodelan Kinematik.....	34
3.5 <i>Flowchart</i> Pemodelan Dinamik.....	34
3.6 <i>Flowchart</i> Pemodelan <i>State Space</i>	35
3.7 <i>Flowchart</i> Perancangan Kontroler.....	36
BAB IV	37
4.1 Hasil Pemodelan Kinematik.....	37
4.2 Hasil Perhitungan Dinamik	38

4.3 Hasil Pemodelan State Space	39
4.4 Perancangan Kontroler Pertama	40
4.5 Perancangan Kontroler Kedua.....	43
4.6 Analisis Respon Pengendali PID dan SMC	45
4.7 Analisa Respon SMC dengan Gangguan.....	48
BAB V	51
5.1 Kesimpulan	51
5.2 Saran	51
DAFTAR PUSTAKA.....	53
LAMPIRAN A	55
LAMPIRAN B	57
LAMPIRAN C	61
LAMPIRAN D	67
LAMPIRAN E	69
LAMPIRAN F.....	77
BIODATA PENULIS.....	79

(Halaman ini sengaja dikosongkan.)

DAFTAR GAMBAR

Gambar 2.1 Perpindahan pada ruang (Tsai,1999).	7
Gambar 2.2 Perpindahan dalam ruang bola (Tsai, 1999).	8
Gambar 2.3 Rotasi pada tiap sumbu tetap (Tsai, 1999).	8
Gambar 2.4 Definisi parameter lengan (Tsai, 1999).	11
Gambar 2. 5 Interpretasi grafik SMC (Holkar, 2013)	26
Gambar 3. 1 <i>Flowchart</i> umum penelitian.....	29
Gambar 3.2 Parameter a_1 dan d_1 pada lengan.....	32
Gambar 3.3 Parameter d_1 pada lengan 1	32
Gambar 3. 4 <i>Flowchart</i> pemodelan kinematik.....	34
Gambar 3. 5 <i>Flowchart</i> pemodelan dinamis	35
Gambar 3. 6 <i>Flowchart</i> pemodelan <i>state space</i>	35
Gambar 3. 7 <i>Flowhart</i> perancangan kontroler.	36
Gambar 4. 1 Skema Diagram Alat.....	37
Gambar 4. 2 <i>Block diagram</i> dari <i>plant</i>	40
Gambar 4. 3 Blok diagram pengendali PID	41
Gambar 4. 4 Blok diagram pengendali <i>sliding mode control</i>	44
Gambar 4. 5 Grafik perbandingan respon gerak azimuth ..	45
Gambar 4. 6 Grafik perbandingan respon gerak elevasi	46
Gambar 4. 7 <i>Block Diagram</i> pengendali SMC dengan penambahan gangguan.....	49
Gambar 4. 8 Grafik respon sudut azimuth dengan penambahan gangguan.....	50
Gambar 4. 9 Grafik respon sudut elevasi dengan penambahan gangguan.....	50

Gambar A 1 Hasil plotting pemodelan kinematis.....	56
Gambar B 1 Matriks A1	57
Gambar B 2 Matriks A2.....	58
Gambar B 3 Matriks A3	59
Gambar B 4 Matriks B	60
Gambar C 1 Langkah-langkah membuat DOE.....	61
Gambar C 2 Halaman <i>Create Response Surface Design</i> . ..	61
Gambar C 3 Langkah-langkah memunculkan halaman <i>Regression</i>	63
Gambar C 4 Halaman Regression	64
Gambar C 5 Hasil analisis regresi untuk respon <i>settling time</i> pada gerak azimuth	64
Gambar C 6 Halaman <i>solver</i> pada Microsoft Excel.....	65
Gambar D 1 Blok diagram kontrol <i>sliding mode control</i> ...	67
Gambar D 2 Subsistem r	68
Gambar D 3 Subsistem s	68
Gambar F 1 Lembar data MAD100B.....	77
Gambar F 2 Kurva karakteristik MAD100B dengan 50 lilitan.	78

DAFTAR TABEL

Tabel 3.1 Parameter Lengan 1 dan 2	32
Tabel 3. 2 Konstanta motor dan <i>gearbox</i> pada blok Simulink.	33
Tabel 3.3 Desain kriteria kontrol alat	33
Tabel 4.1 DH parameter.....	37
Tabel 4. 2 Batasan nilai parameter K_p , K_i dan K_d untuk gerak elevasi.	41
Tabel 4. 3 Batasan nilai parameter K_p , K_i dan K_d untuk gerak elevasi.	42
Tabel 4. 4 Nilai optimal K_p , K_i , dan K_d untuk gerak azimuth. ..	42
Tabel 4. 6 Parameter <i>sliding mode control</i>	44
Tabel 4. 7 Perbandingan performa untuk gerak azimuth	47
Tabel 4. 8 Perbandingan performa untuk gerak elevasi	48
Tabel C 1 DOE <i>Response Surface</i>	62

(Halaman ini sengaja dikosongkan.)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peperangan dapat terjadi kapan saja walaupun sudah merdeka. Perang merupakan pertempuran bersenjata antara dua pasukan atau lebih (KBBI). Perang dapat diartikan sebagai tindakan pemusnahan dan ancaman Negara. Dalam menghadapi sebuah peperangan dibutuhkan strategi dan Alat Utama Sistem Pertahanan (ALUTSISTA) yang kuat. ALUTSISA yang dipakai dalam peperangan modern kali ini banyak berupa kendaraan karena kendaraan memiliki daya hancur yang cukup besar dengan faktor keamanan yang cukup tinggi. Peperangan dapat terjadi di darat, laut dan udara tetapi wilayah darat merupakan tujuan perebutan karena merupakan pusat pemerintahan. Untuk menghadapi peperangan darat maka kendaraan tempur merupakan salah satu ALUTSISTA yang wajib ada dengan teknologi yang mutakhir.

Berdasarkan UU no 3 tahun 2002, pertahanan Negara merupakan segala usaha untuk mempertahankan Negara Kesatuan Republik Indonesia dari ancaman dan gangguan terhadap keutuhan bangsa dan Negara. Dalam UU ini juga disebutkan bahwa sistem pertahanan Negara selayaknya dipersiapkan secara dini oleh pemerintah (pasal1 ayat 2) agar dapat siap siaga dalam menghadapi ancaman bangsa. Seperti yang diketahui, untuk menghadapi ancaman Negara ini dibutuhkan ALUTSISWA yang memadai. Alat perang ini terus berkembang dari zaman ke zaman, dan Indonesia masih menggantungkan teknologinya dengan Negara lain. Hal ini ditunjukkan dengan masih besarnya pembelian armada perang atau senjata ke Negara lain. Kurangnya pengetahuan warga Negara dalam hal pembuatan senjata ini menjadi salah satu faktor ketergantungan Indonesia dalam hal kekuatan ALUTSISTA.

Kebutuhan akan kendaraan tempur menjadi penting karena kendaraan dapat mengurangi tingkat bahaya operatornya.

Pengembangan ALUTSISTA juga menjadi kewajiban Negara, hal ini juga ditunjukkan dalam UU no.3 tahun 2002 bab III pasal 6. Dengan adanya 2 alasan ini, pengembangan pada kendaraan tempurpun diperlukan. Biasanya kendaraan tempur ini dilengkapi dengan *gun turret* atau senapan mesin yang dihubungkan dengan *Remote Control Weapon Station* (RCWS). RCWS ini merupakan salah satu teknologi yang memungkinkan penggunaanya mengoperasikan senapan mesin tanpa harus berada di belakang senapan mesin sehingga menghindarkan operator dari serangan lawan, oleh karena itu dalam penelitian kali ini akan membahas tentang pengembangan RCWS.

Untuk mengembangkan RCWS, dibutuhkan banyak pihak untuk menyelesaikannya mulai dari desain rangka, komponen elektrik termasuk sensor dan pengendaliannya. Untuk penelitian ini hanya difokuskan pada pengendalian dari *Gun turret* dengan ukuran peluru 12,7mm. Sebelum dilakukan pengendalian maka dilakukan dahulu permodelan sistem yaitu dengan menggunakan metode *invers kinematic*. Hal ini dilakukan karena RCWS dianggap sebagai sebuah robot dimana biasanya *Invers kinematic* ini digunakan pada lengan robot. Setelah didapatkan persamaan gerak, langkah selanjutnya adalah melakukan pengendalian.

1.2 Rumusan Masalah

Seperti yang sudah dijelaskan pada latar belakang bahwa pengetahuan tentang ALUTSISTA di Indonesia kurang maka untuk pengembangan RCWS ini cukup menarik. Dimulai dengan mendesain rangka dari RCWS hingga menentukan komponen elektrik, maka dalam bagian pengendalian muncul permasalahan yaitu bagaimana memodelkan sistem dari desain yang telah dibuat dengan menganggap RCWS ini sebagai sebuah robot. Selain itu, permasalahan juga muncul dengan bagaimana pengendalian yang cocok agar RCWS dapat bergerak secara cepat dan tepat.

1.3 Batasan Masalah

Permasalahan yang timbul akan dibahas dengan batasan-batasan yang telah ada. Batasan tersebut adalah sebagai berikut.

- a. Dimensi yang digunakan pada perhitungan diambil dari desain gambar yang telah dibuat.
- b. Senjata yang dipakai adalah *browning* dengan kaliber 12.,7mm
- c. Analisa masa dan geometri berdasarkan gambar yang telah dibuat dan hanya diperhitungkan pada bagian besar.
- d. Pengaruh masa dari motor, *gear box*, dan kopling diabaikan.
- e. RCWS ini terbuat dari bahan *alumunium alloy*.
- f. Perhitungan kinematis digunakan asumsi bahwa RCWS berada tepat diatas tanah.
- g. Pengendalian yang dilakukan hanya pada penembak dan target diam.
- h. Gaya recoil yang timbul sebagai gangguan diabaikan.
- i. Gaya gesekan diabaikan.

1.4 Tujuan Penelitian

Dengan dilakukannya penelitian ini diharapkan beberapa hal agar dapat dicapai yaitu,

- a. Mengetahui persamaan gerakan dari RCWS baik kinematis maupun dinamik.
- b. Mendapatkan pengendalian yang sesuai untuk RCWS.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah

- a. Menambah pengetahuan penulis maupun pembaca dalam hal pembuatan persamaan gerak yang didasarkan pada metode *invers kinematic*.
- b. Sebagai referensi pada penelitian selanjutnya yang akan mengembangkan RCWS dikemudian hari dengan atau tidak menggunakan caliber yang berbeda.
- c. Dapat digunakan sebagai bahan pertimbangan dalam pembuatan RCWS.

(Halaman ini sengaja dikosongkan.)

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Remote Control Weapon Station dapat dikatakan sebagai robot dengan kategori *serial manipulator*. Untuk mendapatkan permodelan kinematis dari sebuah manipulator dapat dilakukan dengan menggunakan metode Lagrangian (Tao, 1999). Penurunan rumus untuk mendapatkan persamaan dinamis menggunakan pengaruh dari inersia tiap komponen. Variabel-variable dan parameter yang telah didapat dari jurnal ini (Tao, 1999) kemudian digunakan untuk mendapatkan permodelan dinamis (Nasyir, 2014 dan Mushonnifah, 2016). Metode Lagrangian juga banyak digunakan untuk mendapatkan persamaan dinamis dengan memanfaatkan penurunan dari energi kinetis. Metode untuk mendapatkan persamaan kinematis lainnya adalah dengan *inverse kinematic* yang sebelumnya ditetapkan matriks transformasi dari setiap pergerakan lengan. Metode perhitungan kinematis dengan menggunakan *invers kinematic* telah digunakan sebagai pendekatan untuk mencari kinematis dari robot 5-DOF (AbuQassem, 2010). Hasil dari kinematis ini juga digunakan untuk mencari matriks Jacobian yang juga mendukung pendekatan kecepatan kinematisnya.

Langkah selanjutnya setelah persamaan dinamis didapatkan adalah mendesain kontroler. Nasyir dalam mengendalikan *Automatic turret gun* menggunakan 3 macam kontroler yaitu PID, *Resolved Acceleration Control* (RAC) dan *esolved Acceleration Control Active Force Control* (RACAFC). RAC digunakan untuk menghilangkan *overshoot* yang terjadi namun eror pada beberapa variasi masih terjadi. RACAFC menghasilkan peforma yang sama dengan RAC yaitu menghilangkan *overshoot*. *Active Force Control* (AFC) juga digunakan Mushonnifah dalam mengkontrol *turret-gun*. Hasil yang didapatkan dengan menggunakan AFC baik karena *error* yang terjadi dibawah 0.1° ketika ingin mencapai

posisi target baik saat diberi gangguan maupun tidak. Berbeda dengan Nasyir dan Mushinnifah, Gang Tao dan Xiaoli Ma (1999), dimana jurnalnya digunakan sebagai referensi, mengkompensasi dari *backlash* dengan solusi berbasis iterasi melalui metode trapezoidal.

PID kontroler merupakan salah satu kontroler tua dan masih dipakai hingga sekarang karena struktur control dari PID itu sederhana, arti fisik dari kontroler ini juga mudah dipahami atau jelas dan sudah banyak operator yang mengenal kontroler ini sehingga menjadi mudah dimplementasikan (Patel, 2012). PID kontroler secara ideal dapat digunakan ketika sistem linear dan tidak bervariasi terhadap waktu (Youney, 2007) dengan kata lain pengendali PID tidak optimal jika digunakan untuk menyelesaikan sistem *nonlinear*.

Pemodelan sistem robot akan menghasilkan sistem *nonlinear* sehingga dibutuhkan pengendali yang mampu menyelesaikan sistem nonlinear yaitu sliding mode control (SMC). Pengendali SMC dipilih oleh beberapa pendahulu untuk mengendalikan sistem nonlinear seperti mengendalikan robot mobil beroda (Ratnawati, 2012) dan inverted pendulum (Nasir, 2010). Pada laporannya, performa pengendali SMC dan PID dibandingkan oleh Nasir dengan sistem inverted pendulum. Performa pengendali PID masih mampu mengendalikan sistem *nonlinear* namun performa pengendali SMC masih lebih baik dibandingkan dengan pengendali PID.

2.2 Permodelan Kinematis

Permodelan kinematis merupakan permodelan dari gerakannya saja tanpa memperhatikan pengaruh dari masa. Permodelan kinematis dapat dilakukan dengan banyak cara namun untuk manipulator robot umumnya menggunakan metode *inverse kinematic*.

2.2.1 Transformasi pada manipulator robot

Awal mula ketika ingin meninjau kinematis dari robot adalah dengan memahami transformasi yang terjadi pada sitiap

lengan. Transformasi ini meliputi posisi dan juga orientasi. Penggabungan dari keduanya merupakan lokasi. Secara lugas untuk membahas transformasi biasanya dibuat pada 4×4 *homogenous transformation matrix*

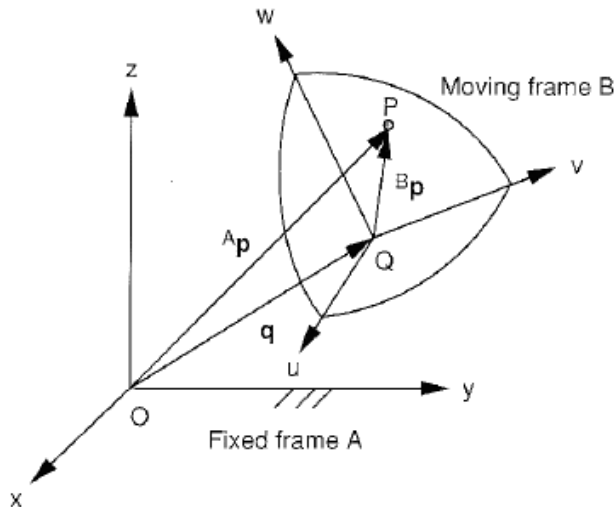
Posisi merupakan titik yang mengacu pada sumbu utama yang dijabarkan dengan matriks 3×1 vektor posisi. Pada gambar 2.1 Ditunjukkan sebuah titik yang mengacu pada *frame A* dan juga mengacu pada *frame B*.

Posisi titik *p* terhadap *frame A* dapat ditulis sebagai berikut

$$p^A = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.1)$$

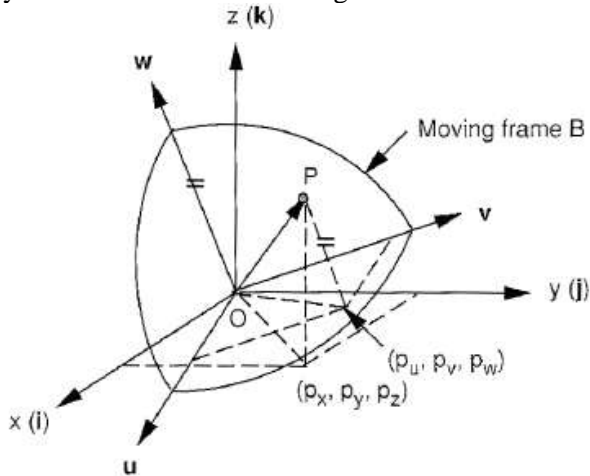
Sedangkan titik *p* terhadap *frame B* dapat ditulis sebagai berikut

$$p^B = \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} \quad (2.2)$$



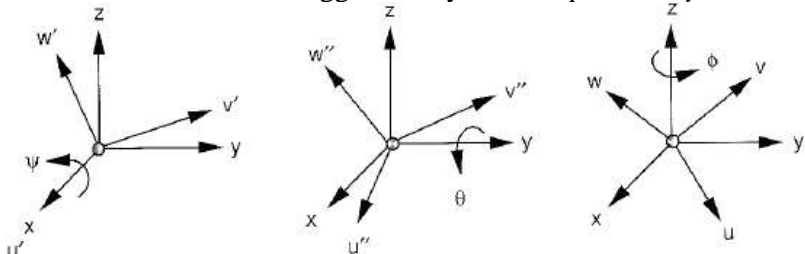
Gambar 2.1 Perpindahan pada ruang (Tsai,1999).

Selain posisi, orientasi juga menjadi bagian dari transformasi. Orientasi dalam hal ini sangat dekat dengan rotasi, dimana rotasi dapat dijelaskan dengan perputaran *frame B* terhadap *frame A* dengan 1 sumbu yang tidak bergerak. Umumnya rotasi ini dinotasikan dengan matriks 3×3 .



Gambar 2.2 Perpindahan dalam ruang bola (Tsai, 1999).

Representasi rotasi yang paling banyak digunakan adalah representasi dari *Euler Angle*. Pada *Euler angle* ini dikenal 3 istilah yaitu *roll*, *pitch* dan *yaw* dimana dapat terlihat pada gambar 2.2. Berurutan dari kiri hingga kanan yaitu *roll*, *pitch* dan *yaw*.



Gambar 2.3 Rotasi pada tiap sumbu tetap (Tsai, 1999).

Rotasi pada sumbu x disebut *roll*, rotasi pada sumbu y disebut *pitch* dan rotasi pada sumbu z disebut dengan *yaw*. Masing-masing rotasi memiliki matriksnya sendiri dan dapat dengan singkat disebut sebagai berikut.

$$\begin{aligned} R_{(x,\psi)} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi & -s\psi \\ 0 & s\psi & c\psi \end{bmatrix} \\ R_{(y,\theta)} &= \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\psi \end{bmatrix} \\ R_{(z,\phi)} &= \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.3)$$

Lokasi merupakan penggabungan penjumlahan dari posisi dan orientasi. Lokasi diperlukan untuk melihat secara keseluruhan perpindahan titik p terhadap titik o (awal). Pada gambar 2.1, lokasi dapat dirumuskan sebagai berikut.

$$\overrightarrow{OP} = \overrightarrow{OQ} + \overrightarrow{QP} \quad (2.4)$$

$$p^A = q^A + R_{(3 \times 3)} \cdot p^B \quad (2.5)$$

Setelah mengetahui lokasi maka selanjutnya *homogenous transformation matrix* harus dicari sebagai semacam peta untuk vector posisi homogen dari satu koordinat ke koordinat lainnya. Matriks ini sebesar 4x4 dan terpisah menjadi 4 bagian. Bagian kiri atas 3x3 matriks merupakan matriks rotasi $R_{(3 \times 3)}$, bagian kanan atas merupakan 3x1 matriks sebagai matriks posisi q^A , bagian kiri bawah 1x3 matriks sebagai *perspective transformation* dan yang terakhir pada kanan bawah 1x1 matriks merupakan *scaling factor*. Untuk manipulator robot 1x3 matriks *perspective transformation* bernilai 0 (Nevatia, 1982). Matrix 4x4 ini dapat dilihat sebagai berikut,

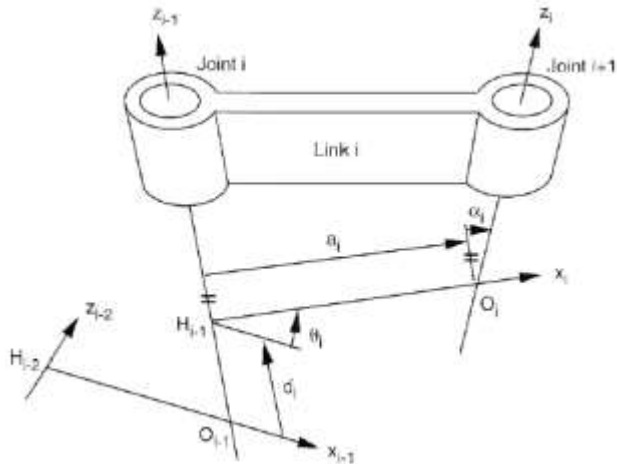
$$T_{(4 \times 4)} = \begin{bmatrix} R_{(3 \times 3)} & q^A \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

2.2.2 Analisa posisi pada Manipulator Seri

Manipulator seri terdiri dari beberapa lengan yang terhubung secara seri dengan bermacam sendi. Karena manipulator ini tersusun secara seri maka terdapat 2 ujung yang tidak terhubung dengan sendi yaitu ujung awal yang terhubung dengan dasar dan ujung lainnya dapat bergerak pada ruang bebas sebagai ujung akhir (Tsai, 1999). Analisa posisi diperlukan karena untuk menentukan performa robot dari masing-masing tugasnya diperlukan lokasi dari *end effector* relative terhadap dasar. Sebelum berlanjut, untuk memahami bagaimana manipulator seri kita harus memahami definisi dari lengan dan parameter yang ada pada manipulator seri.

Pada manipulator seri n -dof terdiri dari lengan awal, n sendi, $n-1$ lengan dan lengan ke n merupakan *end effector* serta merupakan *open loop*. Manipulator ini juga memiliki aktuator pada tiap sendinya. Untuk penomoran, lengan dimulai dari 0 hingga n sedangkan sendi atau *joint* dimulai dari 1 hingga n . telah disebutkan bahwa setiap lengan memiliki 2 sendi kecuali lengan awal atau *base link* dan penggera akhir atau *end effector*. Sendi yang menempel pada awal lengan disebut proximal dan pada akhir disebut distal. Penomoran sumbu dimulai dari 0 hingga n dan terletak pada masing-masing lengan. Pengaturan tiap sumbu dapat dilihat pada gambar 2.4 dan dengan lugas dijelaskan sebagai berikut

- a. z_i -axis berhimpit dengan sumbu sendi ke $(i+1)$
- b. x_i -axis merupakan *common normal* antara sumbu z_{i-1} dengan z_i . Arah dari sumbu x ini dari sendi i ke $(i+1)$. Sumbu x_i juga dapat dibuat sebagai hasil dari *cross product* dari sumbu z_{i-1} dan z_i .
- c. Sumbu y didasarkan pada aturan tangan kanan.
- d. Z_0 berhimpit dengan sendi pertama
- e. X_n digambarkan normal dengan sumbu sendi terakhir $(n-1)$.



Gambar 2.4 Definisi parameter lengan (Tsai, 1999).

2.2.3 Parameter Denavit-Hartenberg (DH Parameter)

Pada penelitian ini digunakan metode Denavit Hartenberg. Parameter dari metode ini bisa didapatkan setelah semua sistem koordinat ditetapkan. Pada 4 parameter pada metode ini yaitu:

- d_i , jarak translasi antara dua garis normal axis yang berpotongan. Jarak tersebut dapat didefinisikan sebagai jarak antara sumbu x_{i-1} ke x_i .
- θ_i , besarnya sudut putar antara axis x_{i-1} menuju x_i pada sumbu z_{i-1} positif sesuai aturan tangan kanan.
- a_i , jarak translasi antara dua sumbu sendi atau *joint axis* yang Secara singkat bisa didapatkan dari jarak antara sumbu z_{i-1} ke z_i .
- α_i , besarnya sudut putar antara axis z_{i-1} menuju z_i pada sumbu x_i positif sesuai aturan tangan kanan.

Setelah masing-masing parameter didapatkan lalu diubahlah menjadi matrix transformasi secara berurutan sebagai berikut

- Perpindahan translasi koordinat sistem $(i - 1)$ sepanjang z_{i-1} sebesar d_i

$$T_{(z,d)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

2. Perpindahan rotasi pada koordinat sistem $(i - 1)$ pada sumbu z_{i-1} dengan sudut θ_i

$$T_{(z,\theta)} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

3. Perpindahan translasi koordinat sistem $(i - 1)$ sepanjang x_i sebesar a_i

$$T_{(z,a)} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

4. Perpindahan rotasi pada koordinat sistem $(i - 1)$ pada sumbu x_i dengan sudut α_i

$$T_{(z,a)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

4 matriks transformasi diatas merupakan transformasi dasar. Untuk mendapatkan matriks transformasi total adalah sebagai berikut.

$${}^{i-1}A_i = T(z, d)T(z, \theta)T(x, a)T(x, \alpha) \quad (2.10)$$

2.2.4 Direct Kinematic

Direct kinematic merupakan salah satu metode dari kinematic robot yang bertujuan untuk mencari vektor posisi q dari *end effector*. *Direct kinematic* dapat digunakan ketika nilai dari masing-masing sudut diketahui. Dengan sendi sejumlah n maka *direct kinematic* dapat dicari melalui persamaan.

$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix} = {}^0A_n \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.11)$$

2.2.5 Inverse Kinematic

Metode lain dari *kinematic* robot merupakan *invers kinematic* dimana metode ini digunakan ketika posisi dan orientasi dari *end effector* diketahui dan digunakan untuk mencari besarnya masing-masing sudut pada *joint*. Setelah mengetahui posisi dan orientasi dari *end effector*, kemudian dibuatlah matriks transformasi *end effector* (Fu, 1987). Secara lebih ringkas dapat dilihat pada persamaan berikut.

$$T_{posisi} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

$$T_{rotasi} = \begin{bmatrix} R(\psi, \theta, \phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

$${}^0T_n = T_{posisi}T_{rotasi} \quad (2.14)$$

Matriks transformasi akhir kemudian dibandingkan dengan matriks transformasi yang telah didapatkan sebelumnya sehingga bisa didapatkan nilai dari masing-masing sudut pada sendi.

2.2.6 Jacobian Matriks

Untuk mengetahui hubungan antara sendi-sendi dengan lokasi *end effector* dapat menganalisa dengan menggunakan *direct kinematic* atau *invers kinematic*. Namun untuk menganalisa kecepatan digunakan matriks Jacobian. Matriks Jacobian merupakan matriks yang merubah kecepatan sendi atau *joint rates* pada actuator menjadi kecepatan pada *end effector* atau secara singkat dapat dirumuskan sebagai

$$\dot{x} = J \dot{q} \quad (2.15)$$

Dimana \dot{q} (*joint space*) merupakan *joint input rates*, \dot{x} (*end effector space*) merupakan *end effector velocities* dan J (Jacobian) adalah transformasi matriks dari *joint input rate* menuju *end effector velocities*. Jacobian dapat mengalami *singularity* ketika dalam kondisi ini, manipulator dapat kehilangan 1 atau lebih derajat kebebasannya.

Perhitungan yang digunakan merupakan *conventional jacobian* dimana semua titik pada *end effector* dapat dijadikan sebagai referensi titik namun biasanya digunakan titik origin. Pada matriks Jacobian ini terdiri dari $2 \times 3 \times n$ submatrices dari matriks Jacobian yaitu J_{vi} disebut *link jacobian matrix* dan J_{wi} disebut *link jacobian submatrices*. Matriks jacobian dapat dengan mudah dipahami melalui persamaan berikut.

$$J_i = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \quad (2.16)$$

$$J_{vi} = \begin{cases} z_{j-1} \times {}^{i-1}P_i & \text{untuk revolute joint} \\ z_{j-1} & \text{untuk prismatic joint} \end{cases}$$

$$J_{wi} = \begin{cases} z_{i-1} & \text{untuk revolute joint} \\ 0 & \text{untuk prismatic joint} \end{cases} \quad (2.17)$$

Untuk menjelaskan kecepatan dari *end effector* dapat dirumuskan sebagai berikut

$$v_n = \sum_{i=1}^n [\dot{\theta}_i (z_{i-1} x^{i-1} p_n^*) + z_{i-1} \dot{d}_i]$$

$$\omega_n = \sum_{i=1}^n \dot{\theta}_i z_{i-1} \quad (2.18)$$

Perhitungan diatas dapat dilakukan dengan mengetahui terlebih dahulu arah dan lokasi dari setiap *joint axis*. Secara berurutan persamaan arah dan lokasi dapat dilihat sebagai berikut

$$z_{i-1} = {}^0R_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.19)$$

$${}^{i-1}p_n = {}^0R_{i-1} {}^{i-1}r_i + {}^i p_n \quad (2.20)$$

2.3 Permodelan Dinamis

Permodelan dinamik merupakan peran penting untuk mencapai *high speed performance* pada sebuah manipulator. Dinamik model dipengaruhi oleh masa atau inersia sebuah benda, momentum, baik linier maupun angular, dan energi kinetic dari sebuah manipulator. Karena permodelan ini dipengaruhi oleh beberapa faktor, maka permodelan dinamik menjadi penting untuk mengetahui pergerakan dari manipulator dengan mendekati kondisi sebenarnya.

2.3.1 Link Inertia Matrices

Inersia dapat dijelaskan sebagai kecenderungan benda untuk diam atau sebagai penghambat dalam perubahan gerakan. Inersia juga dapat dianalogikan sebagai masa namun dalam gerak rotasi. Dalam ruang 3 dimensi, rotasi dapat dilakukan dalam 3 sumbu oleh karena itu besar dari inersia masing-masing sumbu dapat berbeda. Inersia dalam tiga dimensi dapat dituliskan sebagai berikut

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (2.21)$$

I_{xx} merupakan inersia terhadap sumbu x begitu juga dengan I_{yy} dan I_{zz} . I_{xy} merupakan hasil inersia xy begitu juga untuk yang lainnya. Dalam sebuah manipulator dapat terdiri dari beberapa lengan dan dalam setiap lengan dapat terdiri dari beberapa komponen. Untuk mencari inersia dari satu lengan dapat dicari dari inersia masing-masing komponen namun untuk memudahkan, matriks inersia ini juga bisa didapatkan melalui *software design* salah satunya adalah *solidworks*. Dalam perangkat lunak ini terdapat fitur untuk mencari inersia matriks yaitu melalui fitur *mass properties* dimana sebelumnya harus

didefinisikan terlebih dahulu desain secara lengkap (geometri dari masing-masing komponen) dan juga material penyusunnya.

Jika kita meninjau energi kinetic dari sebuah lengan maka pada umumnya inersia matriks (I_i) yang diperhitungkan adalah yang mengacu pada *base frame*. Seperti yang kita ketahui iI_i tidak bergantung terhadap waktu. Namun, (I_i) bergantung pada pose tiap lengan robot. Karena matriks ini mengacu pada koordinat awal dan orientasi dari lengan ke- i terhadap *base* maka dapat dikatakan sebagai fungsi dari *joint variable*. Secara ringkas dapat dirumuskan sebagai berikut

$$I_i = {}^0R_i {}^iI_i ({}^0R_i)^T \quad (2.22)$$

2.3.2 Manipulator Inertia Matrices (M)

Matriks inersia dari sebuah manipulator merupakan total inersia dari sebuah manipulator. Nilainya bisa didapatkan setelah kita mengetahui besarnya inersia matriks dari setiap lengannya terhadap *base* dan matriks Jacobian. Manipulator ini dapat dirumuskan sebagai berikut

$$M = \sum_{i=1}^n (J_{vi}^T m_i J_{vi} + J_{\omega i}^T I_i J_{\omega i}) \quad (2.23)$$

2.3.3 Velocities Coupling Vector (V_i)

Velocities Coupling Vector Merupakan sebuah vektor yang merepresentasikan gaya sentrifugal dari masing-masing lengan. Agar dapat merepresentasikan gaya sentrifugal maka diambil turunan parsial terhadap θ dari inersia matriks manipulator. Agar lebih jelas dapat dirumuskan sebagai berikut.

$$V_i = \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial M_{jk}}{\partial q_i} \right) \dot{q}_j \dot{q}_k \quad (2.24)$$

2.3.4 Gravitational Vector (G_i)

Gravitational vector adalah sebuah vektor yang menunjukkan efek dari adanya percepatan gravitasi. Selain dipengaruhi oleh

gravitasi, vektor ini juga dipengaruhi oleh matriks Jacobian. *Gravitational vector* dapat dirumuskan sebagai

$$G_i = - \sum_{j=1}^n m_j g^T J_{vj}^i \quad (2.25)$$

2.3.5 Formula Lagrangian

Formula Lagrange dapat dicari dengan asumsi tidak ada gaya tambahan pada *end effector* dan tidak adanya gesekan pada sendi. Metode untuk mencari persamaan gerak dari suatu manipulator adalah dengan formula Lagrangian. Metode ini mengeliminasi sebagian atau semua dari batasan pada permulaan. Fungsi Lagrange dapat dirumuskan sebagai selisih antara energi potensial dengan sistem mekanik.

$$L = K - U \quad (2.26)$$

Energi kinetic bergantung pada lokasi serta kecepatan dari lengan sebuah sistem manipulator. Energi potensial hanya bergantung dari lokasi tiap lengan. Perumusan Lagrangian dapat ditulis sebagai berikut.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i, i = 1, 2, \dots, n \quad (2.27)$$

Setelah mendapatkan inersia matriks dari manipulator, *velocities coupling vector* dan vektor gravitasi maka persamaan gerak torsi dari tiap actuator dapat diketahui dan dirumuskan sebagai berikut

$$\sum_{j=1}^n M_{ij} \ddot{q}_j + V_i + G_i = Q_i = \tau_i, i = 1, 2, \dots, n. \quad (2.28)$$

2.4 Pemodelan State Space

Salah satu metode pemodelan adalah dengan menggunakan pendekatan *state*. Pendekatan ini digunakan ketika sistem sudah kompleks dimana dapat memiliki *multiple input* atau *multiple output*. *State space* dapat digunakan untuk memodelkan sistem baik *nonlinear* maupun *linear*, *time invariant* atau *time-*

varying. Pendekatan menggunakan *state space* dilakukan pada *time domain* dengan

State space merupakan *n-dimensional space* dengan axis yang terdiri dari x_1 -axis, x_2 -axis, ..., x_n -axis. Dalam analisisnya digunakan 3 tipe variabel yang mempengaruhi permodelan dari sistem dinamik, yaitu *input variable*, *output variable*, dan *state variable*. (Ogata, 2004). *State variable* merupakan n variabel x_1, x_2, \dots, x_n yang dibutuhkan untuk dapat mewakili keadaan dari sistem dinamik. Jika sistem linear dan tidak bervariasi terhadap waktu maka *state space* dapat dijabarkan dengan n *state variables*, r *input variables* dan m *output variables* dengan koefisien a_{ij}, b_{ij}, c_{ij} , dan d_{ij} adalah konstan maka dapat dituliskan dalam matriks vektor berikut

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{2.29}$$

dimana

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \mathbf{B} \\ &= \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nr} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix} s \\ \mathbf{y} &= \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}, \mathbf{D} \\ &= \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1r} \\ d_{21} & d_{22} & \cdots & d_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mr} \end{bmatrix}\end{aligned}$$

Matriks **A**, **B**, **C** dan **D** secara berurutan merupakan *state matrix*, *input matrix*, *output matrix* dan *direct transmission*

matrix. Vektor \mathbf{x} , \mathbf{u} dan \mathbf{y} merupakan *state vector*, *input vector* dan *output vector* (Ogata, 2004).

Model matematis dari persamaan (2.28) tidak dapat digunakan langsung untuk dikontrol oleh karena itu harus diubah menjadi *state space* model. Model terlebih dahulu harus dilinierkan dan diubah dengan menggunakan Matriks Jacobian. Berikut adalah Matriks Jacobian untuk masing-masing matriks A, B, C dan D (Gajic, 2003).

$$A^{n \times n} = \frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \quad (2.30)$$

$$B^{n \times r} = \frac{\partial f}{\partial u} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_r} \\ \frac{\partial f_2}{\partial u_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \dots & \frac{\partial f_n}{\partial u_r} \end{pmatrix} \quad (2.31)$$

$$C^{p \times n} = \frac{\partial y}{\partial x} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial y_p}{\partial x_1} & \dots & \dots & \frac{\partial y_p}{\partial x_n} \end{pmatrix} \quad (2.32)$$

$$D^{pxr} = \frac{\partial y}{\partial u} = \begin{pmatrix} \frac{\partial y_1}{\partial u_1} & \frac{\partial y_1}{\partial u_2} & \dots & \frac{\partial y_1}{\partial u_r} \\ \frac{\partial y_2}{\partial u_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial y_p}{\partial u_1} & \dots & \dots & \frac{\partial y_p}{\partial u_r} \end{pmatrix} \quad (2.33)$$

2.5 Pengendalian

2.5.1 Time Response

Setelah didapatkan permodelan secara matematis dari sebuah siste maka hal selanjutnya adalah menganalisa *transient response* dari sebuah sistem. Analisa dari respon transien dapat diketahui melalui beberapa karakteristik dari respon tersebut (Nise,2011). Dilihat dari sistem, sistem ini merupakan orde 2. Secara umum sistem orde 2 mengikuti persamaan berikut.

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.33)$$

Frekuensi natural (ω_n) pada sistem orde 2 merupakan frekuensi dari osilasi sistem tanpa adanya pengaruh dari *damping*. Secara umum frekuensi natural dapat dirumuskan sebagai $\omega_n = \sqrt{k/m}$. Sedangkan *damping ratio* (ζ) merupakan perbandingan *exponential decay frequency* terhadap frekuensi natural dari sebuah sistem. Melalui persamaan 2.30, *damping ratio* dapat dicari dengan persamaan dengan a merupakan koefisien dari s pada persamaan 2.29

$$\zeta = \frac{a}{2\omega_n} \quad (2.34)$$

Terdapat 4 karakteristik yang dapat dilihat untuk mengetahui respon transien yaitu

- a. *Rise time*, adalah waktu yang dibutuhkan sistem dari 0.1 nilai akhir hingga mencapai 0.9 dari nilai akhir.
- b. *Peak time*, adalah waktu yang dicapai saat nilai maksimum. Nilai ini dapat dicari dengan menggunakan persamaan

$$T_p = \frac{\pi}{\sqrt{1 - \zeta^2}} \quad (2.35)$$

- c. *Percent overshoot*, %OS, adalah nilai prosentase dari nilai maksimum terhadap *final value*. Besarnya prosentase dapat dicari dengan perumusan sebagai berikut.

$$\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100\% \quad (2.36)$$

Dari persamaan %OS juga dapat mencari nilai *damping ratio* yaitu dengan mengikuti persamaan berikut.

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \quad (2.37)$$

- d. *Settling Time*, merupakan waktu yang dibutuhkan untuk osilasi sistem memasuki daerah $\pm 2\%$ dari *steady state value*. Nilai ini dapat dicari dengan persamaan berikut

$$T_s = \frac{4}{\zeta\omega_n} \quad (2.38)$$

2.5.2 Desain Kontroler PID

Ketika sebuah sistem dijalankan akan terjadi keadaan yang tidak diinginkan dari sebuah sistem seperti adanya *steady state error* atau sistem tidak memenuhi kriteria dari *time response*. Untuk mengatasi itu maka diberilah kompensator. Pada dasarnya pemberian kompensator yang dimaksud adalah dengan menambahkan *poles* dan atau *zeros* pada sistem.

Untuk memperbaiki *steady-state error* dapat menggunakan kompensator yang menambahkan *pole* pada *origin* dan *zeros* yang terletak dekat dengan *origin*. Letak pasti dari *zeros* nantinya akan dihitung melalui cara-cara *root locus*. Hal ini dapat memperbaiki *steady-state error* karena akan menaikkan sejumlah satu tipe dari

sistem, kompensator ini dinamakan *ideal integral compensator* atau *PI controller*. Dalam pengaplikasiannya, kontroler ini membutuhkan integrator aktif. Jika kita ingin menggunakan pasif sirkuit maka letak *pole* dan *zero* akan bergeser ke kiri tetapi tetap mendekati *origin*. Perbaikan dari *steady-state error* akan terlihat dengan perbandingan z_c terhadap p_c dimana nilainya sama dengan perbaikan *static error constant* K_v dari sebelum dan setelah ada kompensator. Besar perbaikan pada K_v akan mempengaruhi juga besarnya perbaikan pada *steady-state error*. Kompensator ini dinamakan dengan *lag compensation*. Untuk lebih jelasnya *PI* dan *lag compensation* dapat dilihat dari persamaan berikut.

$$PI ; K \frac{s + z_c}{s} \quad (2.39)$$

$$lag ; K \frac{(s + z_c)}{(s + p_c)} \quad (2.40)$$

Respon transien tidak sesuai dengan yang kita harapkan maka kita dapat menambahkan kompensator untuk memenuhi keinginan kita. Kompensator ini bekerja dengan menambahkan 1 *zero* pada sistem. Peletakan *zeros* ini menyesuaikan dengan *transient response specification*. Kompensator dinamakan dengan *ideal derivative compensation* atau *PD controller*. Sama halnya dengan *PI*, *PD* juga membutuhkan sirkuit aktif dan untuk menggunakan sirkuit pasif maka *pole* dan *zero* akan bergeser ke kiri. Kelebihan dengan menggunakan sirkuit pasif adalah tidak perlu adanya tambahan *power supplies*, *noise* dari adanya diferensiasi berkurang namun kekurangannya adalah kompensator ini (*lead compensation*) tidak dapat mereduksi *branch* dari *root locus* yang melewati sumbu imajiner sedangkan *PD* bisa. Untuk lebih jelasnya *PD* dan *lead compensation* dapat dirumuskan sebagai berikut.

$$PD ; K(s + z_c) \quad (2.41)$$

$$Lead ; K \frac{(s + z_c)}{(s + p_c)} \quad (2.42)$$

Untuk mendapatkan keuntungan dari kedua kompensator yang telah dijelaskan maka digunakanlah *PID controller*. Pada

umumnya untuk mendesain *PID controller* terlebih dahulu kita memperbaiki respon transien yang menggunakan *PD* kompensator kemudian barulah kita tambahkan *Integral* kompensator untuk memperbaiki *steady-state response*. *Transfer function* dari *PID* kompensator ini adalah

$$G_c(s) = K_1 + \frac{K_2}{s} + K_3 s = \frac{K_1 s + K_2 + K_3 s^2}{s} \quad (2.43)$$

PID controller juga membutuhkan sirkuit aktif, maka untuk menggantinya dengan sirkuit pasif kita harus mendesain *lag-lead compensator*. Kedua kompensator ini dapat yaitu

$$PID ; k \frac{(s + z_{lag})(s + z_{lead})}{s} \quad (2.45)$$

$$Lag - lead ; k \frac{(s + z_{lag})(s + z_{lead})}{(s + p_{lag})(s + p_{lead})} \quad (2.46)$$

2.5.3 Rancangan Percobaan *Response Surface*

Suatu rancangan percobaan diperlukan untuk melakukan analisis regresi agar pengambilan data dilakukan secara sistematis. Metode *response surface* adalah salah satu rancangan percobaan yang umum digunakan. Keunggulan metode ini adalah minimnya pengambilan data yang digunakan dan keluaran model dapat berbentuk kuadratik (orde dua) dengan regresi. Nilai rancangan eksperimen berbentuk *coded*, dengan berurutan angka 1, -1 dan 0 menunjukkan nilai maksimum, minimum dan nilai tengah. Angka 1,682 dan -1,682 menunjukkan titik aksial dan respon ditunjukkan dengan Y. Secara umum, bentuk rancangan eksperimen *response surface coded* dapat dilihat pada tabel 2.1. (Montgomery, 2001)

Tabel 2. 1 Rancangan eksperimen *response surface*

No.	A	B	C	Y
1	-1	-1	-1	Y1
2	1	-1	-1	Y2

3	-1	1	-1	Y3
4	1	1	-1	Y4
5	-1	-1	1	Y5
6	1	-1	1	Y6
7	-1	1	1	Y7
8	1	1	1	Y8
9	-1,682	0	0	Y9
10	1,682	0	0	Y10
11	0	-1,682	0	Y11
12	0	1,682	0	Y12
13	0	0	-1,682	Y13
14	0	0	1,682	Y14
15	0	0	0	Y15
16	0	0	0	Y16
17	0	0	0	Y17

2.5.3 Analisis Regresi

Analisis regresi bertujuan untuk memperoleh model matematis berdasarkan data yang dimiliki. Perlakuan yang tepat bagi proses dapat ditentukan berdasarkan model matematis yang telah diperoleh. Microsoft Excel dan Minitab adalah perangkat lunak yang dapat digunakan untuk membantu proses analisis tersebut. Akan tetapi, Minitab memiliki superioritas dalam pengolahan data statistik.

Salah satu kegunaan dari analisis regresi adalah menentukan model matematis gaya dari suatu proses pemesinan (Chua, 1993). Analisis regresi yang digunakan adalah regresi kuadratik (model orde 2). Bentuk taksiran orde kedua untuk 3 variabel bebas adalah:

$$\begin{aligned}\hat{Y} = & b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_1X_2 \\ & + b_5X_1X_3 + b_6X_2X_3 + b_7X_1^2 \\ & + b_8X_2^2 + b_9X_3^2\end{aligned}\quad (2.47)$$

dengan:

\hat{Y}	= Nilai taksiran
b_o	= Konstanta
b_i	= Taksiran parameter
X_i	= Variabel bebas

2.5.4 Optimasi GRG Non-linier

Generalized Reduced Gradient (GRG) merupakan salah satu algoritma optimasi untuk persamaan non-linier (Lasdon, 1975). GRG dapat digunakan untuk mengoptimalkan suatu persamaan non-linier yang berkelanjutan (*continuous*). GRG terdiri dari satu fungsi tujuan dan beberapa fungsi batasan (*constrain*). Fungsi tujuan dapat berupa meminimalkan, memaksimalkan, ataupun mencapai suatu nilai tertentu. Fungsi batasan bertujuan untuk menentukan daerah dimana nilai tujuan dapat dicapai.

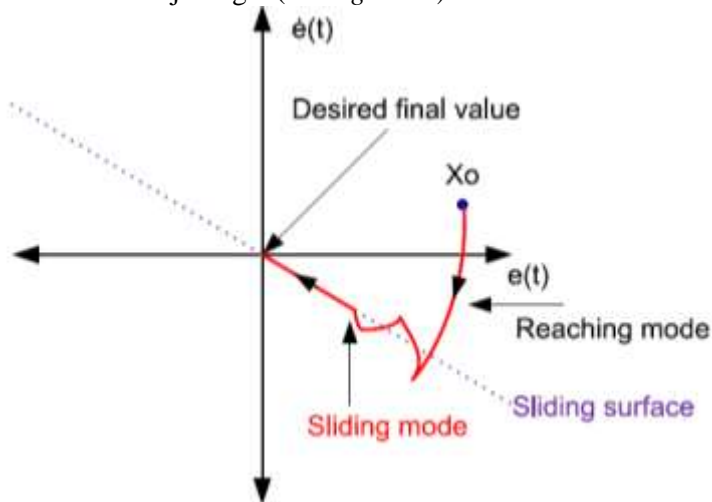
GRG telah berkembang dan dapat ditemukan pada *solver* Microsoft Excel (Harmon, 2011). Persamaan yang diperoleh dari regresi mula-mula dimasukan pada Microsoft Excel. Batasan-batasan yang digunakan, seperti nilai maksimum ataupun nilai minimum dari suatu variabel, dapat dimasukan pula pada Microsoft Excel. Setelah semua data telah diberikan, dapat ditentukan fungsi tujuan dan batasan pada *solver*. Selain itu, perlu juga ditentukan tujuan dari optimasi yang dilakukan, seperti meminimalkan atau memaksimalkan suatu nilai. Metode GRG kemudian dipilih. *Solver* Microsoft Excel akan memberitahu apabila ditemukan suatu nilai yang optimal. Nilai tersebut optimal hanya untuk *range* yang telah ditentukan.

2.5.5 Sliding Mode Control

Sliding mode control (SMC) merupakan sebuah teori control yang menggunakan pendekatan *state space* untuk menganalisa sebuah sistem (Nasir, 2010). Penggunaan *state space* ini baik digunakan karena dapat merepresentasikan model yang kompleks serta sistem MIMO. Metode SMC ini dikembangkan untuk

membangun sistem control pada model yang kompleks dengan sistem dinamik nonlinear derajat tinggi yang bekerja dalam keadaan tidak tentu (Rahmani, 2016).

Cara kerja SMC dalam mengendalikan sebuah sistem adalah dengan menjalankan sistem pada sebuah permukaan yang kemudian nilainya dijaga pada permukaan ini. Gambar 2.5 merepresentasikan cara kerja SMC dengan menggunakan fase bidang yang terdiri dari eror dan turunan pertama dari eror tersebut. Dapat dilihat bahwa pada saat dimulai dari berbagai kondisi, keadaan trayektori akan menuju permukaan *sliding* (*reaching mode*), dan kemudian akan meluncur sepanjang permukaan menuju target (*sliding mode*).



Gambar 2. 5 Interpretasi grafik SMC (Holkar, 2013)

Langkah pertama dalam membangun SMC adalah dengan menentukan persamaan permukaan (*sliding surface*). Persamaan *sliding surface* akan mengikuti persamaan 2.56, dimana hanya dipengaruhi oleh 1 (satu) parameter skalar yaitu λ yang ditentukan oleh desainer dan bernilai positif konstan (Slotine, 1991). n merupakan besarnya order dari suatu sistem.

$$S(t) = \left(\lambda + \frac{d}{dt} \right)^{n-1} e(t) \quad (2.48)$$

Teorema kestabilan Lyapunov digunakan untuk mestabilan suatu sistem nonlinier yang berubah terhadap waktu. Fungsi Lyapunov digunakan dan dapat dilihat pada permaan 2.49. Persamaan tersebut dikatakan sebuah fungsi definit positif apabila memenuhi syarat $V(0) = 0, V > 0$.

$$V = \frac{1}{2} S^2 \quad (2.49)$$

Teorema 2.1 (Rahmani, 2016), menganggap bahwa sistem didefinisikan sebagai $\dot{x} = f(x, t)$ dengan $f(0, t) = 0$ untuk semua t . Jika fungsi scalar yang terdapat pada $V(x, t)$ bersifat kontinu, maka turunan parsial pertama memenuhi kondisi $\dot{V}(x, t)$ definit negatif. Dalam kondisi ini, sistem mengalami *uniformly asymptotic stable*. Berdasarkan Teorema 2.1 yang diterapkan pada persamaan 2.49, persamaan 2.50 memenuhi syarat kestabilan sistem yang bersifat definit negatif. Nilai ε adalah suatu konstanta positif .

$$\dot{V} = s\dot{s} \leq -\varepsilon |s| < 0 \quad (2.50)$$

Dengan fungsi signum (sgn) dapat dirumuskan pada persamaan 2.51 atau dapat didefinisikan sebagai berikut.

$$\text{sgn}(x) \begin{cases} -1, & x < 1 \\ 1, & x \geq 1 \end{cases} \quad \text{sgn}(x) = \frac{|x|}{x} \quad (2.51)$$

Dengan definisi dari fungsi signum, persamaan 2.50 dapat dituliskan dalam bentuk lain seperti pada persamaan 2.51 yang dapat disebut juga sebagai *Constant Rate Reaching Law* (Liu, 2012).

$$\dot{s} = -\varepsilon \text{sgn}(s), \varepsilon > 0 \quad (2.52)$$

Membangun kontroler dengan metode *sliding mode control* juga melibatkan model sistem yang akan dikontrol. Representasi dari model sistem sesuai pada persamaan 2.29.

$$\ddot{\theta}(t) = Ax(t) + Bu(t) \quad (2.53)$$

Persamaan *sliding surface* merupakan fungsi dari eror. Dengan nilai orde $n=2$, persamaan 2.58 dapat disederhanakan sebagai berikut.

$$S(t) = \dot{e}(t) + \lambda e(t) \quad (2.54)$$

Fungsi eror dan turunannya adalah sebagai berikut dengan r adalah posisi ideal.

$$\begin{aligned} e(t) &= r(t) - \theta(t), \dot{e}(t) = \dot{r}(t) - \dot{\theta}(t), \ddot{e}(t) \\ &= \ddot{r}(t) - \ddot{\theta}(t) \end{aligned} \quad (2.55)$$

Dengan mendapati turunan pertama dari persamaan 2.54 dan memasukan persamaan 2.53. dan 2.55 maka didapatkan persamaan berikut

$$\dot{S} = \ddot{r}(t) - A\ddot{x}(t) - B\ddot{u}(t) + \lambda(\dot{r}(t) - \dot{\theta}(t)) \quad (2.56)$$

Berlakuknya *constant rate reaching law*, maka persamaan 2.56 dapat disamakan dengan persamaan 2.52 sehingga didapatkan persamaan berikut

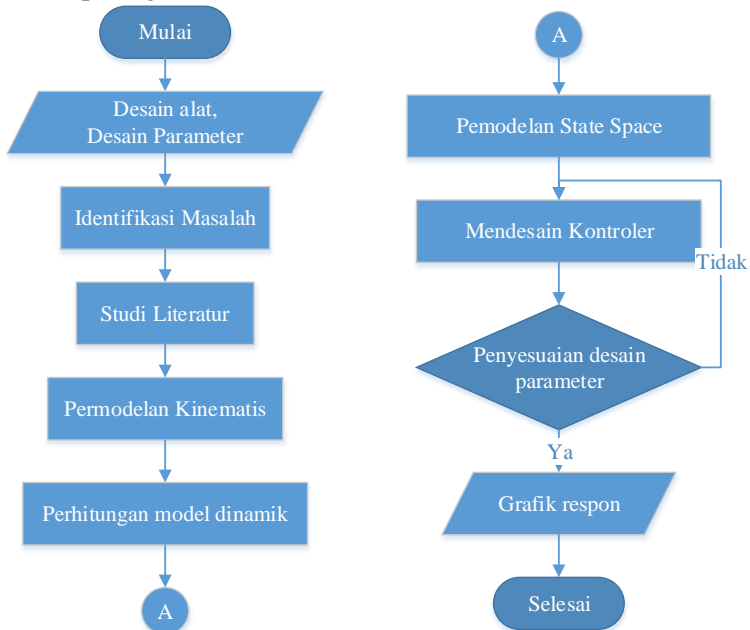
$$\begin{aligned} u(t) &= \frac{1}{B} \left(\varepsilon \operatorname{sgn}(s) + \ddot{r}(t) - A\ddot{x}(t) \right. \\ &\quad \left. + \lambda(\dot{r}(t) - \dot{\theta}(t)) \right) \end{aligned} \quad (2.57)$$

BAB III

METODOLOGI

3.1 Flowchart Penelitian

Penelitian ini dilakukan dengan kerja sama beberapa tim. Masing-masing tim memiliki tugasnya masing-masing. Penelitian ini bergerak di dalam tim kontrol. Pada tim ini melakukan kegiatan sebagai pembuat sistem kontrol mulai dari permodelan kinematis, dinamis hingga pembuatan pengendaliannya. Namun sebelum itu dilakukan beberapa pengujian tentang parameter disain terhadap gaya statis. Parameter desain atau desain ini telah dibuat oleh tim desain. Diagram alur untuk keseluruhan tim dapat dilihat pada gambar 3.1.



Gambar 3. 1 Flowchart umum penelitian

3.2 Metode Penelitian

Dalam menentukan permodelan dinamis dan juga mendesain *controller*, akan dilakukan beberapa tahapan yang akan dilakukan, yaitu:

1. Studi literatur.

Untuk memperdalam pemahaman mengenai sistem yang akan dianalisa, dilakukan studi literature yang berkaitan dengan manipulator seri dengan dua *revolute joint* baik secara permodelan kinematis hingga permodelan dinamis, mendesain *controller PID* dan *sliding mode control*. Studi literature ini diperoleh dari buku-buku teks, journal, *e-book* dan penelitian terdahulu.

2. Penetapan DH parameter alat.

DH parameter pada alat dilakukan sebelum tahap permodelan kinematis dan dinamis. Karena sistem yang ditinjau merupakan sebuah manipulator robot, maka parameter ini dibutuhkan sebagai representasi dari sistem robot ini. Parameter-parameter ini mengikuti aturan Denavit-Hartenberg serta menghasilkan matriks transformasi untuk mengetahui pergerakan *end effector*.

3. Perhitungan model kinematis

Permodelan kinematis ini dimulai dengan mencari matriks transformasi dari masing-masing sendi. Selanjutnya matriks transformasi dari end effector terhadap base ditetapkan, kemudian kedua matriks transformasi ini disamakan. Metode ini dinamakan dengan invers kinematic. Hasil dari invers kinematic adalah persamaan sudut dari sistem, persamaan ini juga dapat dikatakan sebagai persamaan kinematis. Untuk menentukan besarnya kecepatan sudut dan percepatan sudut maka dibutuhkan matriks Jacobian. Untuk mempercepat perhitungan, digunakan *software Matlab*.

4. Perhitungan model dinamis

Perhitungan model dinamis baru dapat dilaksanakan ketika sudah mendapatkan nilai dari massa tiap lengan beserta matriks

inersia tiap lengan yang kemudian akan digunakan untuk memperhitungkan manipulator inersia matriks. Permodelan dinamis didapatkan dengan formulasi Lagrangian dimana memperhitungkan velocities coupling vector dan gravitational vector. Hasil dari formulasi ini adalah torsi sebagai fungsi dari sudut, kecepatan sudut dan percepatan sudut. Perhitungan model dinamis menggunakan software Matlab untuk mempercepat perhitungan.

5. Pemodelan *State Space*

Pemodelan dinamis akan digunakan untuk memodelkan sistem dengan menggunakan pendekatan *state space*. Pendekatan ini dilakukan karena model bersifat Multi Input Multi Output (MIMO). Perubahan model dari persamaan gerak menjadi *state space model* dapat dilakukan dengan menggunakan matriks Jacobian seperti pada persamaan (2.30 – 2.33)

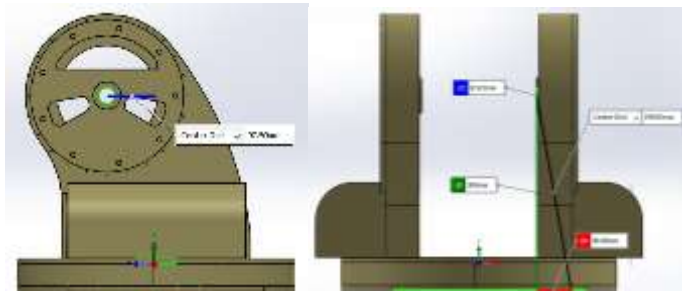
6. Mendesain Kontroler.

Kontroler yang digunakan dalam penelitian kali ini terdapat 2 yaitu kontroler dengan menggunakan PID dan Sliding Mode Control. Kedua kontroler ini kemudian dibandingkan untuk mendapatkan yang terbaik. Mendesain kontroler PID tidak lepas dari menentukan nilai dari konstanta KP, KI, dan KD. Penentuan nilai-nilai ini dilakukan metode optimasi stokastik, yaitu GRG Nonlinear. Sliding mode control memiliki 2 konstanta yang harus ditetapkan untuk masing-masing kontroler (kontroler sudut1 dan sudut2). Penentuan konstanta ini dilakukan dengan cara *trial and error* hingga respon memenuhi kriteria desain.

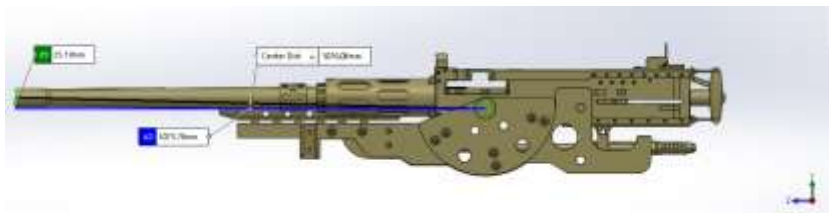
3.2 Parameter RCWS 12.7 mm

3.3.1 Dimensi dan Desain RCWS 12.7 mm

Pemodelan sistem RCWS 12.7mm dilakukan berdasarkan desain sesuai gambar 3.2 dan 3.3. Desain gambar RCWS 12.7 mm merupakan desain RCWS ITS (Wahyudi, 2016). Matriks inersia masing-masing lengan diperoleh melalui perangkat lunak Solidworks. Parameter desain tiap lengan telah dirangkum pada tabel 3.1.



Gambar 3.2 Parameter a_1 dan d_1 pada lengan



Gambar 3.3 Parameter d_1 pada lengan 1

Tabel 3.1 Parameter Lengan 1 dan 2

Parameter	Ukuran
Panjang a lengan 1	97.5 mm
Panjang d lengan 1	389 mm
Panjang d lengan 2	1075.78 mm
Matriks inersia lengan 1	$\begin{bmatrix} 1 & -0.02 & -0.01 \\ -0.02 & -0.63 & -0.78 \\ 0.01 & 0.78 & -0.63 \end{bmatrix} \text{ Kgm}^2$
Matriks inersia lengan 2	$\begin{bmatrix} 0.00 & 0.00 & 1.00 \\ -0.07 & -1.00 & 0.00 \\ 1.00 & -0.07 & 0.00 \end{bmatrix} \text{ Kgm}^2$

3.3.2 Desain Motor dan Gearbox

Sumber gerakan tiap lengan berasal dari motor listrik yang dikopel dengan *gearbox* untuk mereduksi keluaran motor. Motor listrik dan *gearbox* dianggap ideal dengan secara efektif memberikan nilai yang konstan langsung setelah dinyalakan. Motor yang digunakan adalah Motor dari Rexroth IndraDyn A dengan tipe Mad 100B dan lilitan sebanyak 50. Lembar Data untuk motor ini dapat dilihat pada lampiran F. Nilai konstanta motor yang digunakan adalah nilai torsi konstan pada 20°C. Konstanta untuk masing-masing motor dan *gearbox* dirangkum pada tabel 3.2

Tabel 3. 2 Konstanta motor dan *gearbox* pada blok Simulink.

Konstanta	Gerak Azimuth	Gerak Elevasi
Motor listrik (Nm/A)	7,66	7,66
<i>Gearbox</i>	16,67	16,67

3.3.3 Desain Kriteria Kontrol Alat

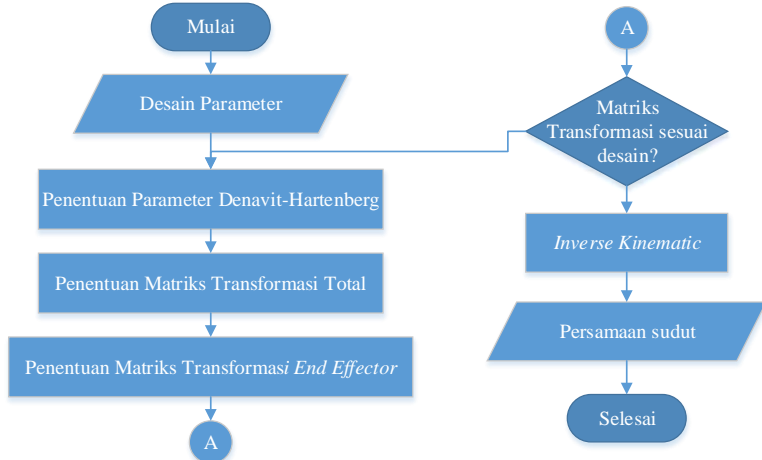
Dalam mendesain sebuah kontroler diperlukan parameter tujuan yaitu harapan sistem ini berjalan. Parameter control ini adalah kecepatan maksimum dari masing-masing pergerakan dapat dilihat pada tabel 3.4

Tabel 3.3 Desain kriteria kontrol alat

Parameter	Gerak Azimuth	Gerak Elevasi
<i>Settling time (Ts)</i>	5 sekon	5 sekon
% <i>Overshoot (%OS)</i>	<5%	<5%
<i>Root Mean Square Steady State Error</i>	<2%	<2%

3.4 Flowchart Pemodelan Kinematik

Pada permodelan kinematik akan dilakukan beberapa tahapan yang bertujuan untuk mendapatkan persamaan tiap sudut sendi dan matriks transformasi total dari desain parameter RCWS 12.7 mm. Tahapan-tahapan dapat dengan jelas dilihat pada gambar 3.5.



Gambar 3. 4 Flowchart pemodelan kinematik

3.5 Flowchart Pemodelan Dinamik

Beberapa tahap diperlukan untuk melakukan permodelan dinamik. Masukkan untuk tahap ini berupa matriks transformasi dari tahap sebelumnya dan matriks inersia yang didapat dari perangkat lunak. Permodelan dinamik dilakukan bertujuan untuk mendapatkan persamaan gerak nonlinier dari sistem. Tahap-tahap yang akan dilakukan dapat dilihat pada gambar 3.5.



Gambar 3. 5 *Flowchart* pemodelan dinamis

3.6 *Flowchart* Pemodelan *State Space*

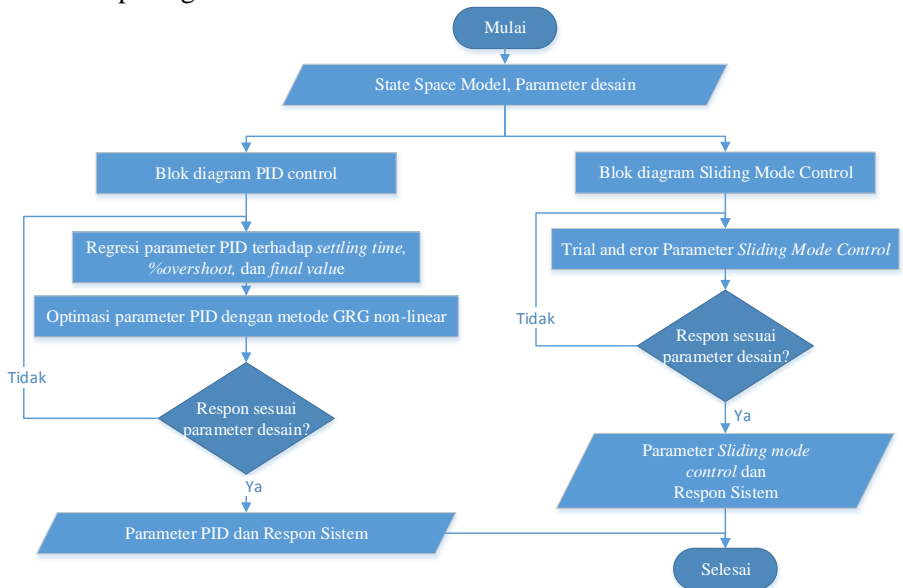
Pada tahap ini dilakukan beberapa tahap yang bertujuan untuk mendapatkan model *state space* dengan masukan berupa persamaan gerak nonlinear dari tahap sebelumnya. Tahap-tahap itu dapat dilihat pada gambar 3.6.



Gambar 3. 6 *Flowchart* pemodelan *state space*.

3.7 Flowchart Perancangan Kontroler

Pada tahap ini dibutuhkan model state space dari tahap sebelumnya untuk dikontrol. Perancangan pengendalian menggunakan 2 tipe kontroler yaitu PID dan sliding mode control dimana keduanya memiliki parameter masing. Tahapan-tahapan untuk menentukan parameter dan merancang kontroler dapat dilihat pada gambar 3.7.

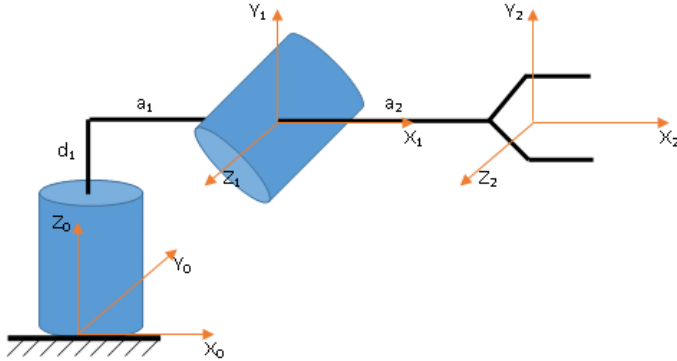


Gambar 3. 7 Flowhart perancangan kontroler.

BAB IV

ANALISA HASIL DAN PEMBAHASAN

4.1 Hasil Pemodelan Kinematik



Gambar 4. 1 Skema Diagram Alat

Manipulator yang digunakan merupakan manipulator seri dengan 2 lengan. Manipulator seri ini dapat dilihat secara skema pada gambar 4.1. Parameter DH ditetapkan berdasarkan subbab 2.2.3 dan dapat dilihat pada tabel 4.1.

Tabel 4.1 DH parameter

Joint _i	d _i	Θ _i	a _i	α _i
1	0.389 m	Θ ₁	0.0975 m	270°
2	0 m	Θ ₂	1.07578 m	0°

Setelah penetapan parameter DH maka matriks transformasi total bisa diperoleh dan dilihat pada persamaan persamaan 4.1. Setelah matriks transformasi total didapatkan maka dapat dilakukan *inverse kinematic* untuk memperoleh persamaan sudut untuk masing-masing sendi seperti pada persamaan 4.2.

$${}^0A_2 = \begin{bmatrix} \cos(\theta_1) \cos(\theta_2) & -\cos(\theta_1) \sin(\theta_2) & -\sin(\theta_1) & 0.0975 \cos(\theta_1) + \sin(\theta_1) \sin(\theta_2) \\ \cos(\theta_2) \sin(\theta_1) & -\sin(\theta_1) \sin(\theta_2) & \cos(\theta_1) & 0.0975 \sin(\theta_1) + \sin(\theta_1) \cos(\theta_2) \\ -\sin(\theta_1) & -\cos(\theta_2) & 0 & 0.389 - 1.076 \sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$\begin{aligned}
\theta_1 = & -\log(((- 5.0e95 * \exp(teta2 * 2.0i) - 2.5e95 \\
& * \exp(teta2 * 4.0i) + 1.0e96 * \exp(teta2 \\
& * 2.0i) * \cos(psi)^2 * \cos(teta)^2 \\
& - 2.5e95)^{(1/2)} + 1.0e48 * \exp(teta2 \\
& * 1.0i) * \cos(psi) * \cos(teta)) / (5.0e47 \\
& * \exp(teta2 * 2.0i) + 5.0e47)) * 1.0i \\
\theta_2 = & -1.0 * \operatorname{asin}(1.0 * \sin(psi))
\end{aligned} \tag{4.2}$$

4.2 Hasil Perhitungan Dinamik

Perhitungan dinamik membutuhkan matriks transformasi sebagai masukannya. Setelah matriks ini didapatkan maka dapat melakukan perhitungan matriks inersia dan matriks jacobian tiap lengan. Setelah kedua matriks didapatkan maka dapat melakukan perhitungan Lagrange dengan terlebih dahulu mendapatkan matriks inersia sistem, vektor kecepatan kopel dan vektor gravitasi. Persamaan nonlinear keluaran dari perhitungan Lagrange ini digunakan pada penelitian ini agar mendekati keadaan sebenarnya. Hasil yang didapatkan dapat dilihat pada persamaan 4.3.

$$\begin{aligned}
Torsi1 = & 0.541 * tetadotdot1 - 1.0 * tetadot2^2 \\
& * \cos(teta2) + 0.918 * tetadotdot1 \\
& * \cos(2.0 * teta2) - 0.035 * tetadotdot1 \\
& * \sin(2.0 * teta2) + 1.03 * tetadotdot1 \\
& * \cos(teta2) - 1.0 * tetadotdot2 \\
& * \sin(teta2) - 1.03 * tetadot1 * tetadot2 \\
& * \sin(teta2) - 0.07 * tetadot1 * tetadot2 \\
& * \cos(2.0 * teta2) - 1.84 * tetadot1 \\
& * tetadot2 * \sin(2.0 * teta2) \\
Torsi2 = & 2.84 * tetadotdot2 - 51.7 * \cos(teta2) + 0.514 \\
& * tetadot1^2 * \sin(teta2) + 0.035 \\
& * tetadot1^2 * \cos(2.0 * teta2) + 0.07 \\
& * tetadotdot1 * \cos(teta2) + 0.918 \\
& * tetadot1^2 * \sin(2.0 * teta2) - 1.0 \\
& * tetadotdot1 * \sin(teta2) - 0.035 \\
& * tetadot1 * tetadot2 * \sin(teta2)
\end{aligned} \tag{4.3}$$

4.3 Hasil Pemodelan State Space

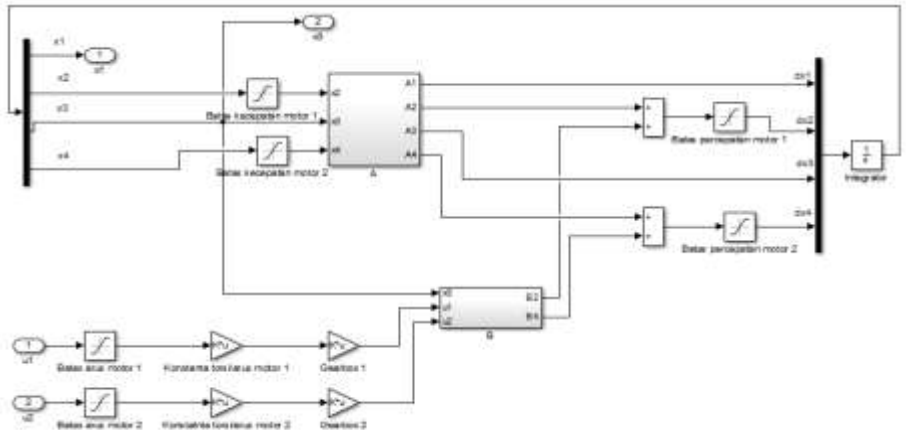
Pemodelan *state space* membutuhkan pentetapan *state variable* terlebih dahulu. Setelah ditetapkan, penyederhanaan dan perhitungan dari persamaan nonlinear dilakukan untuk mendapatkan *state equation*. Sevara berurutan *state variable* dan *state equation* dapat dilihat pada persamaan 4.4 dan 4.5.

$$\begin{aligned}x_1 &= \theta_1 \\x_2 &= \dot{\theta}_1 \\x_3 &= \theta_2 \\x_4 &= \dot{\theta}_2\end{aligned}\tag{4.4}$$

$$\begin{aligned}\dot{x}_1 &= \dot{\theta}_1 \\x_2 &= (5.0e66 * u1 + 9.11e67 * \cos(x3) * \sin(x3) \\&\quad + 1.76e66 * u2 * \sin(x3)) / (4.59e66 \\&\quad * \cos(2.0 * x3) - 1.75e65 * \sin(2.0 * x3) \\&\quad + 5.14e66 * \cos(x3) + 1.23e65 * \cos(x3) \\&\quad * \sin(x3) - 1.76e66 * \sin(x3)^2 \\&\quad + 2.7e66) \\\dot{x}_3 &= \dot{\theta}_2 \\x_4 &= (0.353 * u2 + 18.2 * \cos(x3) - (0.0247 * u1 \\&\quad * \cos(x3)) / (0.918 * \cos(2.0 * x3) - 0.035 \\&\quad * \sin(2.0 * x3) + 1.03 * \cos(x3) \\&\quad + 0.541) + (0.353 * u1 * \sin(x3)) / (0.918 \\&\quad * \cos(2.0 * x3) - 0.035 * \sin(2.0 * x3) \\&\quad + 1.03 * \cos(x3) + 0.541)) / ((0.0247 \\&\quad * \cos(x3) * \sin(x3)) / (0.918 * \cos(2.0 \\&\quad * x3) - 0.035 * \sin(2.0 * x3) + 1.03 \\&\quad * \cos(x3) + 0.541) - (0.353 \\&\quad * \sin(x3)^2) / (0.918 * \cos(2.0 * x3) \\&\quad - 0.035 * \sin(2.0 * x3) + 1.03 * \cos(x3) \\&\quad + 0.541) + 1.0)\end{aligned}\tag{4.5}$$

State equation kemudian diubah menjadi *state space* dengan menggunakan matriks Jacobian sehingga didapatkan nilai matriks A, B, C dan D. pemodelan matriks A dan B dapat dilihat pada lampiran B. *State space* yang diperoleh merupakan persamaan nonlinear sehingga perangkat lunak Matlab tidak dapat

mengubahnya menggunakan *syntax* yang tersedia. Pemodelan dilakukan secara manual dengan menggunakan Simulink. Permodelan *state space* pada Simulink dapat dilihat pada gambar 4.2



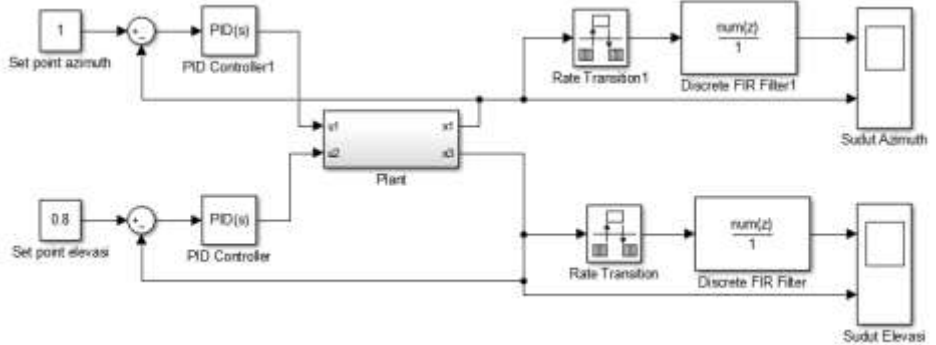
Gambar 4. 2 Block diagram dari plant.

Pada kenyataannya, motor dan arus listrik yang terjadi memiliki batasan-batasan tertentu sehingga dalam simulasi model digunakan blok *saturation* untuk membatasi nilai-nilai tersebut. Selain member batasan, pada simulasi ini juga sudah diberikan konstanta untuk spesifikasi motor dan pengaruh reduksi dari *gearbox* untuk masing-masing lengan. Konstanta ini diwakilkan dengan blok *gain*. Nilai dari spesifikasi motor merupakan perbandingan antara torsi motor yang dihasilkan dengan arus yang dibutuhkan sebuah motor.

4.4 Perancangan Kontroler Pertama

Kontroler pertama menggunakan metode PID. Dalam simulasinya, pengendali pid menggunakan blok yang telah tersedia dalam program Simulink sehingga dapat langsung digunakan. Susunan kontroler PID dapat dilihat pada gambar 4.3. Pemilihan parameter PID menggunakan optimasi GRG nonlinear

sehingga membutuhkan data untuk dianalisa. Hasil keluaran sistem beresilasi, hal ini dikarenakan sistem memperhitungkan model dinamik, digunakan filter *moving average* untuk mempermudah menganalisa sistem. Filter yang digunakan membutuhkan data diskrit sehingga dibutuhkan blok *rate transition* untuk mengubah data dari kontinu menjadi diskrit.



Gambar 4. 3 Blok diagram pengendali PID

Pengendali PID memiliki 3 parameter yang harus diatur, yaitu K_p , K_i , dan K_d . Batasan nilai dari ketiga parameter ini ditentukan berdasarkan hasil simulasi. Batasan-batasan tersebut dirangkum pada tabel 4.2 dan 4.3.

Tabel 4. 2 Batasan nilai parameter K_p , K_i dan K_d untuk gerak elevasi.

Parameter	Nilai Minimum	Nilai Maksimum
K_p	30	70
K_i	20	40
K_d	1	5

Tabel 4. 3 Batasan nilai parameter Kp, Ki dan Kd untuk gerak elevasi.

Parameter	Nilai Minimum	Nilai Maksimum
Kp	20	39
Ki	20	40
Kd	5	10

Nilai dari ketiga parameter ini digunakan sebagai faktor dari sistem dan nilai *settling time*, *percent overshoot*, dan *final value* digunakan sebagai respon. Suatu rancangan percobaan dibentuk dengan menggunakan metode *response surface*. Dipilih *design of experiment* (DOE) *response surface* karena jumlah data yang dibutuhkan seminimal mungkin namun dapat membentuk model kuadratik (orde 2) dengan analisis regresi. Eksperimen dilakukan menurut desain eksperimen agar sistematis dan diambil nilai dari respon. Setelah data-data hasil eksperimen diambil, dilakukan analisa regresi dan menghasilkan persamaan kuadratik. Cara pengerjaan regresi untuk masing-masing respon beserta hasilnya dapat dilihat pada lampiran C.

Persamaan hasil regresi digunakan untuk mendapatkan nilai optimal masing-masing parameter dengan menggunakan metode optimasi GRG non-linear. Pada *solver* Microsoft Excel, fungsi tujuan dan batasan dimasukkan. Parameter pada fungsi tujuan juga didefinisikan. Nilai-nilai optimal dari masing-masing parameter diperoleh dan digunakan. Nilai-nilai tersebut dapat dilihat pada tabel 4.4.

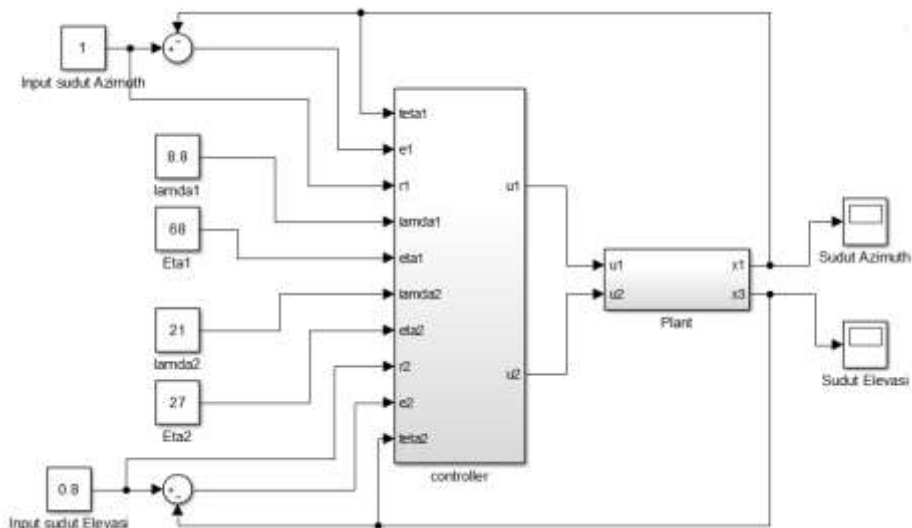
Tabel 4. 4 Nilai optimal Kp, Ki, dan Kd untuk gerak azimuth.

Parameter	Nilai Optimal Gerak Azimuth	Nilai Optimal Gerak Elevasi
Kp	37,98	34,98
Ki	40	38,66
Kd	5	5,85

4.5 Perancangan Kontroler Kedua

Pada mulanya metode kontrol kedua dirancang menggunakan metode *predictive control* dengan memakai blok yang sudah ada pada *library* Simulink. Variasi terhadap *predictive horizon* dan *control horizon* dilakukan, tetapi respon kedua sudut tidak dapat mencapai *set point*. Metode ini tidak dapat digunakan karena terdapat eror pada filter Kalman. Dugaan pertama muncul dari sistem nonlinear yang digunakan. Deret Taylor digunakan untuk melinearkan sistem. Hasil linearisasi sistem ini tidak memberikan hasil yang baik saat dilakukan pengendalian. *Steady state error* yang muncul dari pengendalian ini cukup besar sehingga dipilih metode control lain. Dokumentasi penelitian dengan metode *predictive control* dapat dilihat pada lampiran.

Pengendalian dengan *sliding mode control* (SMC) dipilih karena sederhana untuk digunakan dan dapat mengakomodasi sistem yang kompleks, bahkan sistem nonlinear. Blok diagram SMC dapat dilihat pada gambar 4.4. Pembuatan blok diagram kontrol SMC mengikuti persamaan 2.57.



Gambar 4. 4 Blok diagram pengendali *sliding mode control*.

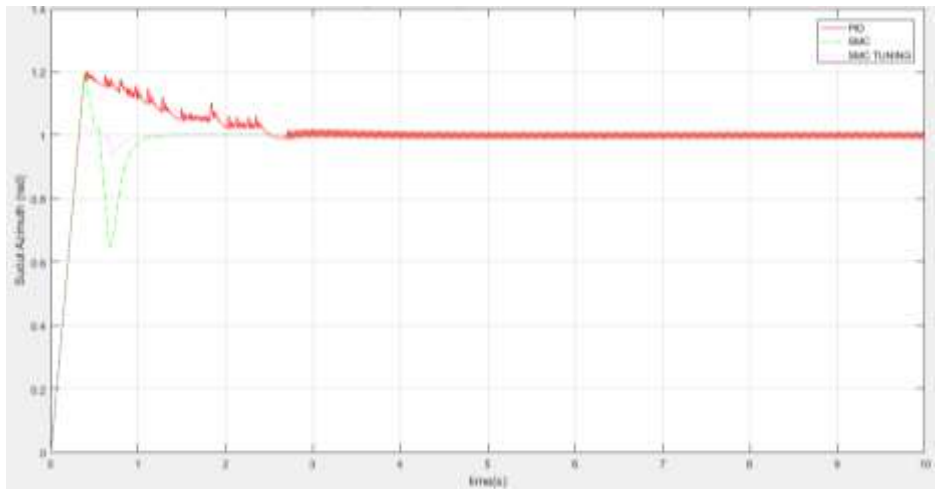
Pada gambar 4.4 terlihat terdapat parameter untuk menentukan seberapa besar kemiringan *sliding surface*. Parameter ini dicari dengan cara *trial and error* dengan *range* yang ditentukan. Respon sistem yang tidak stabil, *percent overshoot* yang besar dan *settling time* lama menjadi batasan dalam menentukan *range* nilai parameter. Nilai optimal yang diperoleh dari masing-masing parameter dapat dilihat pada tabel 4.6

Tabel 4. 5 Parameter *sliding mode control*

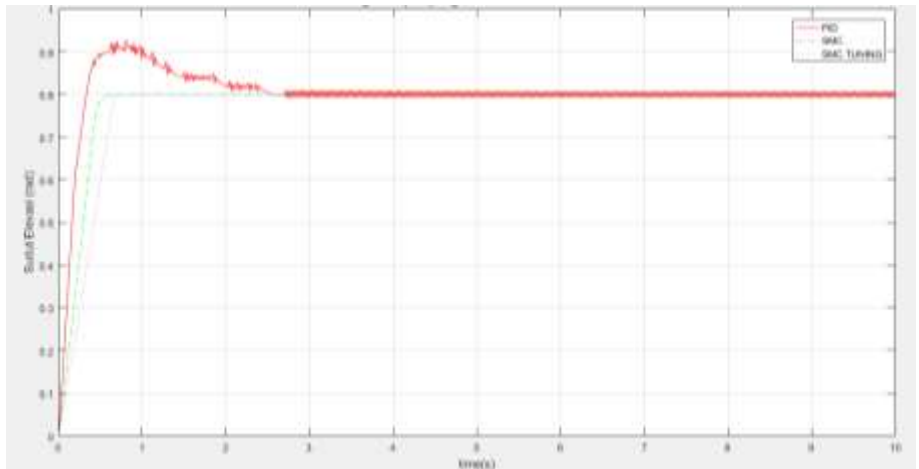
Parameter	Nilai Konstanta	Nilai Tuning
Lamda1	9.7	6.8
Eta1	78	128
Lamda2	19	83
Eta2	31	97

4.6 Analisis Respon Pengendali PID dan SMC

Nilai optimal dari masing-masing pengendali dimasukkan dan responnya dibandingkan untuk masing-masing gerak azimuth dan elevasi. Perbandingan ini dilakukan untuk mencari pengendali yang lebih baik. Secara berurutan perbandingan respon untuk gerakan azimuth dan elevasi dapat dilihat pada gambar 4.5 dan 4.6. Pada kedua gambar itu, respon oleh pengendali PID ditunjukkan dengan garis merah, PID hasil *tuning* dengan garis biru putus-putus, pengendali SMC dengan garis *dash-dot* hijau dan SMC hasil *tuning* dengan garis titik-titik ungu. Ketiga performa yang dievaluasi, *settling time* (T_s), % *Overshoot* (%OS) dan *Root Mean Square SteadyState Error* (RMS SS Error), dikatakan semakin baik apabila nilainya semakin kecil.



Gambar 4. 5 Grafik perbandingan respon gerak azimuth



Gambar 4. 6 Grafik perbandingan respon gerak elevasi

Terlihat respon pengendali PID mengalami osilasi dikarenakan perhitungan dinamikanya.

Hasil respon PID memiliki nilai RMS SS Error yang mendekati 0. Nilai *overshoot* dari respon mendekati batas $\pm 20\%$ dari nilai referensi. Pada pengendali PID, baik gerakan azimuth ataupun elevasi, terlihat respon pada keadaan transien mengalami osilasi begitu cepat dan tidak mulus, hal ini mengakibatkan gerakan yang tidak mulus atau bergetar. pada saat keadaan *steady*, juga terjadi *chattering* yang mengakibatkan gerakan senjata trus bergetar sehingga dapat mengakibatkan menurunnya ketepatan dalam menembak.

Tuning diperlukan pengendali SMC karena *overshoot* dan *undershoot* terjadi cukup besar yaitu mendekati batas $\pm 20\%$ dari nilai referensi. Secara kasat mata perbaikan besar dari sistem ada pada nilai *overshoot* gerak azimuth. Jika pada SMC untuk gerakan Azimuth, nilai *overshoot* sebesar 19,88% sedangkan pada *tuning* nilainya hanya 3,65%. Nilai dari T_s pada gerak azimuth sebesar 0.857, tetapi T_s gerak elevasi meningkat sebesar 0.1 detik. RMS SS Error yang terjadi mengalami penurunan. Pada

gerak azimuth hanya terjadi 0.003 rad dan gerak elevasi 0.002 rad.

Respon pengendali SMC tidak terjadi osilasi. Keunggulan utama SMC terletak pada kecepatan respon dalam mencapai *settling time* (Ts). Keunggulan pada Ts ditunjukkan pada kedua gerak (azimuth dan elevasi) dengan nilai Ts kendali PID melebihi 2 kali nilai Ts SMC. Performa %OS pada pengendalian SMC tidak terlalu baik, karena nilainya mendekati batas maksimal %OS yang diijinkan namun hal ini masih unggul dibandingkan pengendali PID yang melebihi batas. Pada pengendali SMC timbul *undershoot* yang besarnya hamper sama dengan %OS. Gerak elevasi dengan pengendali SMC hamper tidak memiliki %OS yaitu hanya sebesar 0.5%. Nilai *RMS SS Error* baik pada gerak azimuth maupun elevasi, pengendalian SMC lebih baik 2x dibandingkan dengan pengendali PID. Secara keseluruhan, nilai perbandingan performa antara kedua pengendali dapat dilihat pada tabel 4.6 untuk gerak azimuth dan tabel 4.7 untuk gerak elevasi. Menurut teori, pengendali SMC akan memiliki potensi terjadi *chattering* lebih besar tetapi pada sistem ini tidak terjadi. Hal ini dapat terjadi karena tidak semua sistem akan terjadi *chattering* yang besar. Jika grafik respon diperbesar, *chattering* masih akan terlihat tetapi dengan jarak nilai maksimum dan minimum kecil sehingga tidak terlihat kasat mata.

Tabel 4. 6 Perbandingan performa untuk gerak azimuth

Parameter performa	PID	SMC	SMC <i>Tuning</i>
<i>Settling time</i>	2,554 sekon	0,96 sekon	0.857 sekon
<i>% Overshoot</i>	22,7 %	19,88%	3.65%
<i>RMS SS Error</i> (rad)	0.009	0.009	0.003

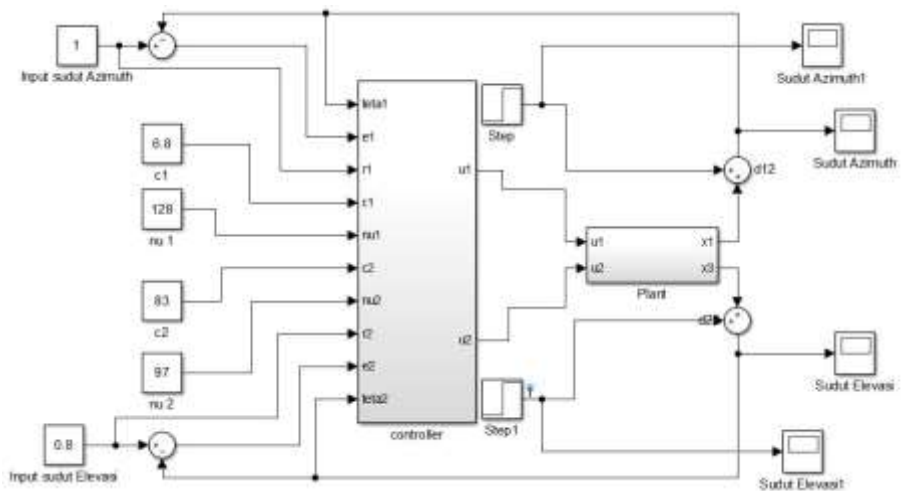
Tabel 4. 7 Perbandingan performa untuk gerak elevasi

Parameter performa	PID	SMC	SMC <i>Tuning</i>
<i>Settling time</i>	1,878 sekon	0,5 sekon	0.6 sekon
<i>% Overshoot</i>	15,83 %	0.5%	0.01%
<i>RMS SS Error</i> (rad)	0.0055	0.005	0.002

Dari hasil pengamatan performa, masing-masing pengendali PID dan SMC dapat mengendalikan sistem nonlinear. Pengendali PID dan SMC setelah *tuning* menghasilkan respon dengan RMS SS *Error* yang sangat kecil. Namun, pengendali SMC menghasilkan respon yang lebih baik dibandingkan dengan pengendali PID dari performa *settling time* dan *overshoot*.. Pada mulanya pengendali SMC menghasilkan nilai *overshoot* yang mendekati batas, namun setelah mengalami *tuning*, nilai *overshoot* menurun. Setelah di-*tuning*, pengendali SMC dapat memberikan respon yang baik. Berdasarkan evaluasi hasil performa masing-masing pengendali, maka pengndali SMC dipilih sebagai pengendali RCWS 12,7mm agar memiliki respon yang baik.

4.7 Analisa Respon SMC dengan Gangguan.

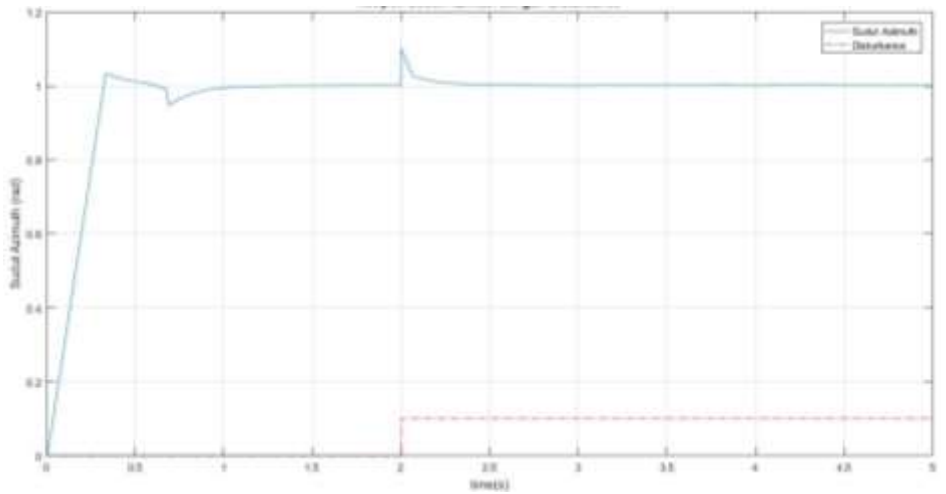
Sliding Mode Control (SMC) merupakan suatu metode pengendalian yang tahan terhadap gangguan, oleh karena itu akan ditambahkan gangguan. Gangguan yang ditambahkan adalah gangguan luar seperti angin dan di wakilkkan dengan *step input*. Gangguan luar diletakkan dengan menambahkan nilainya setelah keluar dari *plant*. Nilai gangguan luar yang diberikan sebesar 10% dari nilai *set point*. *Block diagram* setelah diberikan gangguan dapat dilihat pada gambar 4.7.



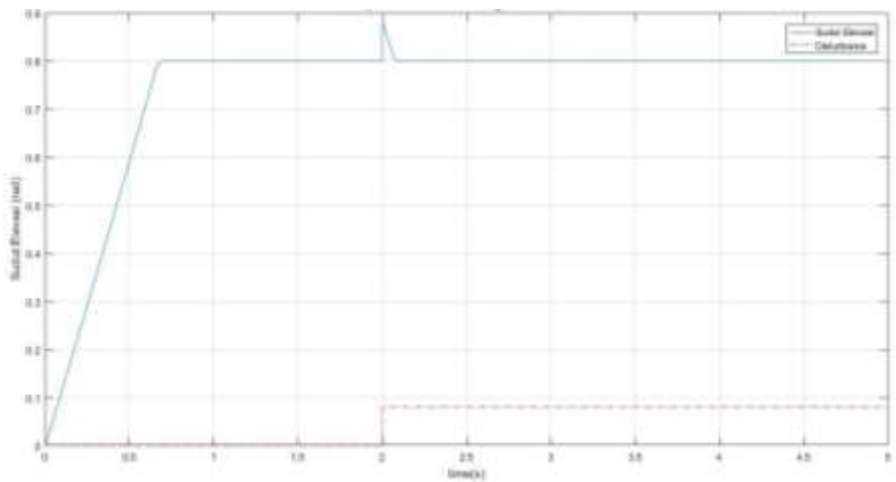
Gambar 4. 7 Block Diagram pengendali SMC dengan penambahan gangguan

Seperti yang tertera pada subbab sebelumnya, *settling time* paling lama terjadi pada gerak azimuth dengan lama 0.8 sekon maka nilai gangguan ditambahkan setelah respon berada dalam keadaan *steady*. Pada simulasi ini penambahan gangguan dilakukan pada detik ke 2. Grafik respon dan gangguan untuk masing-masing gerakan dapat dilihat pada gambar 4.8 dan 4.9.

Pada masing-masing grafik terdapat 2 garis yaitu garis biru menunjukkan respon an garis merah putus-putus merupakan representasi dari gangguan dimana pada detik ke 2 gangguan baru saja ditambahkan. Jika diamati dengan baik, saat detik ke 2 respon mengalami peningkatan akibat gangguan, tetapi tidak mencapai 0.2 sekon respon sudah kembali ke keadan *steady*. Hal ini membuktikan pengendali tahan terhadap gangguan atau *robust*.



Gambar 4. 8 Grafik respon sudut azimuth dengan penambahan gangguan.



Gambar 4. 9 Grafik respon sudut elevasi dengan penambahan gangguan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini dilakukan untuk mendapatkan permodelan dan pengendalian yang sesuai untuk RCWS 12,7mm. Kesimpulan yang dapat diperoleh dari penelitian ini adalah sebagai berikut,

1. Diperoleh model kinematis dari RCWS 12,7mm yaitu sesuai pada persamaan 4.2.
2. Diperoleh model dinamis dari RCWS 12,7mm yaitu sesuai pada persamaan 4.3.
3. Dipilih pengendali *sliding mode control*, dengan konstanta untuk gerak azimuth yaitu λ_1 dan ν_1 sebesar 6,8 dan 128 sedangkan konstanta untuk gerak elevasi yaitu λ_2 dan ν_2 sebesar 83 dan 97. Konstanta yang telah dipilih dan digunakan akan menghasilkan performa pada gerak azimuth dengan nilai *settling time* sebesar 0.857 sekon, %*Overshoot* sebesar 3,65% dan *Root Mean Square Steady State Error* sebesar 0,003rad dan performa pada gerak elevasi ialah, nilai *settling time* sebesar 0,5 sekon, %*Overshoot* sebesar 0,01% dan *Root Mean Square Steady State Error* sebesar 0,002rad.
4. Pengendali SMC terbukti bersifat *robust* setelah ditambahkan gangguan eksternal.

5.2 Saran

Penelitian ini masih memiliki banyak kekurangan, sehingga beberapa saran yang dapat dilakukan adalah sebagai berikut

1. Penelitian selanjutnya dapat mempertimbangkan gesekan yang terjadi. Apabila terjadi kesulitan untuk memperhitungkan gesekan, dapat menggunakan metode *system identification* ketika alat sudah jadi.
2. *Sliding mode control* semakin populer karena kelebihanannya, hal ini diikuti juga dengan pengembangan-pengembangan dari metode ini. Untuk penelitian berikutnya bisa juga menggunakan pengembangan dari SMC.

DAFTAR PUSTAKA

- AbuQassem, M. R. (2010), "*Simulation Interfacing of 5 DOF Educational Robot Arm*", Thesis: Electrical Engineering Department, Islamic University of Gaza.
- Fu, K. S., Gonzales, R. C. dan Lee, C.S.G. (1987), "*Robotics: Control, Sensing, Vision, and Intelligent*", McGraw-Hill, New York.
- Gajic, Zoran. (2003), "*Linear Dynamic Systems and Signals*", New Jerseys: Pretince Hall.
- Holkar, K. S. (2013), "*Sliding Mode Control with Predictive PID Sliding Surface for Improved Performance*", International Journal of Computer Application, Vol 78- No.4.
- Liu, Jinkun dan Wang, Xinhua. (2012), "*Advanced Sliding Mode Control for Mechanical Systems*", Tsinghua University Press, Beijing and Springer-Verlag Berlin Heidelberg 2012.
- Mushonnifah, Siti. 2016."Resolved Acceleration Control (RAC) DAN Active Force Control (AFC) pada Sistem Turret-Gun Kaliber 20 milimeter". Surabaya: Jurusan Matematika, Institut Teknologi Sepuluh Nopember.
- Nasir, A. N. K. Ismail, R. M. T. Raja. dan Ahmad, M. A. (2010), "*Performance Comparison between Sliding Mode Control (SMC) and PD-PID Controllers for a Nonlinear Inverted Pendulum System*", International Journal of Computer, Electricalm Automation, Control and Information Engineering Vol:4, no:10.
- Nasyir, M. T. (2014), "*Aplikasi Kontrol Cerdas pada Automatic Gun Turret Sebagai Penolakan Gangguan Di Kendaraan Tempur Darat*", Thesis: Jurusan Teknik Mesin, Institut Teknologi Sepuluh Nopember, Surabaya.
- Nise, N. S. (2011), "*Control System Engineering sixth edition*", John Wiley & Sons, Inc, Jefferson City.
- Ogata, K. (2010), "*Modern Control Engineering*". New Jersey: Prentice Hall.

- Slotine, Jacques E. dan Li, W. (1991), "*Applied Nonlinear Control*". New Jersey: Prentice Hall.
- Patel H. B. (2012), "*Developments in PID Controllers: Literature Survey*", IJEIRS.
- Subiono (2013), "Sistem Linear dan Kontrol Optimal", Subiono, Jurusan Matematika, Institut Teknologi Sepuluh Nopember, Surabaya.
- Rahmani. Mehran, Ghanbari. Ahmad, dan Ettafagh. Mir Mohammad. (2016), "Robust adaptive control of a bio-inspired robor manipulator using bat algorithm", Expert Systems with Applications 56 (2016) 164-176.
- Ratnawati (2012), "DESAIN PENGENDALIAN ROBOT MOBIL BERODA MENGGUNAKAN METODE *SLIDING MODE CONTROL* (SMC)", Tugas Akhir Jurusan Matematika, Institut Teknologi Sepuluh Nopember, Surabaya.
- Tao, Gong, dan Ma, Xiaoli. (1999), "*Backlash Compensation for Multivariable Nonlinear System with Actuator Dynamics*", IEEE.
- Tsai, Lung-Wen, (1999), "*Robot Analysis The Mechanical of Serial and Parallel Manipulators*", John Wiley & Sons, Inc, Canada.
- Youney, Justin, (2007), "*A Comparison And Evaluation of Common Pid Tuning Methods*", Thesis: University of Central Florida, STARS.

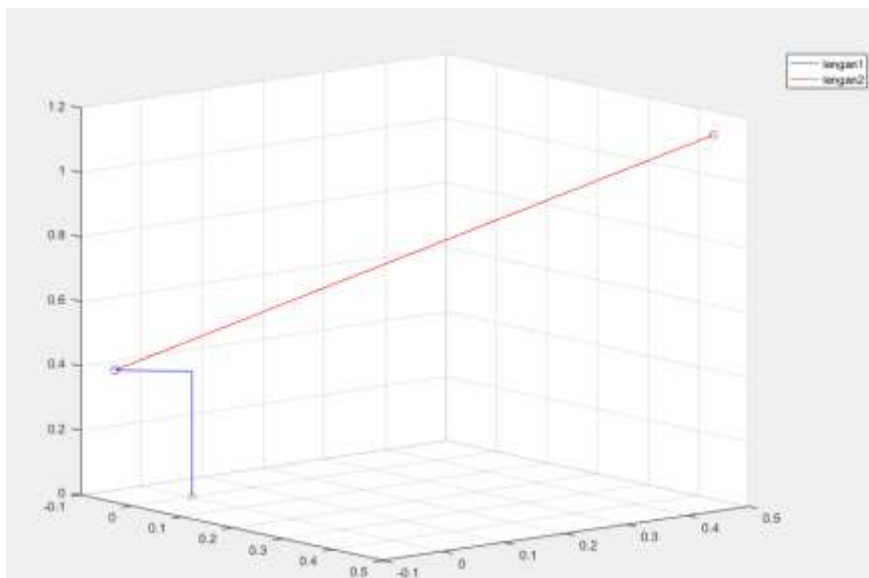
LAMPIRAN A

PLOTING MODEL KINEMATIS

Pada lampiran ini akan ditampilkan plot model kinematis. Plotting ini akan digunakan untuk memverifikasi hasil permodelan kinematis. Plotting dilakukan dengan bantuan perangkat lunak Matlab. Data yang dibutuhkan adalah posisi dari setiap pergerakan yang diambil dari matriks transformasi. Hasil yang tampak dapat dilihat pada gambar A1. Lengan 1 ditunjukkan dengan garis warna biru sedangkan lengan 2 ditunjukkan dengan garis warna merah. Berikut adalah *coding* untuk menghasilkan verifikasi hasil pemodelan kinematis.

Koding tambahan untuk plotting hasil pemodelan kinematis.

```
pos(1:3,1) = [0;0;0];  
pos(1:3,2) = A0(1:3,4);  
pos(1:3,3) = A1(1:3,4);  
pos(1:3,4) = TDH(1:3,4);  
  
%%ploting  
hold on  
for i = 1 : (size(pos,2)-1)  
    plot3([pos(1,i) pos(1,i+1)], [pos(2,i)  
    pos(2,i+1)], [pos(3,i) pos(3,i+1)], 'r')  
end  
plot3(pos(1,1), pos(2,1), pos(3,1), 'o');  
plot3(pos(1,3), pos(2,3), pos(3,3), 'o');  
plot3(pos(1,4), pos(2,4), pos(3,4), 'o');
```

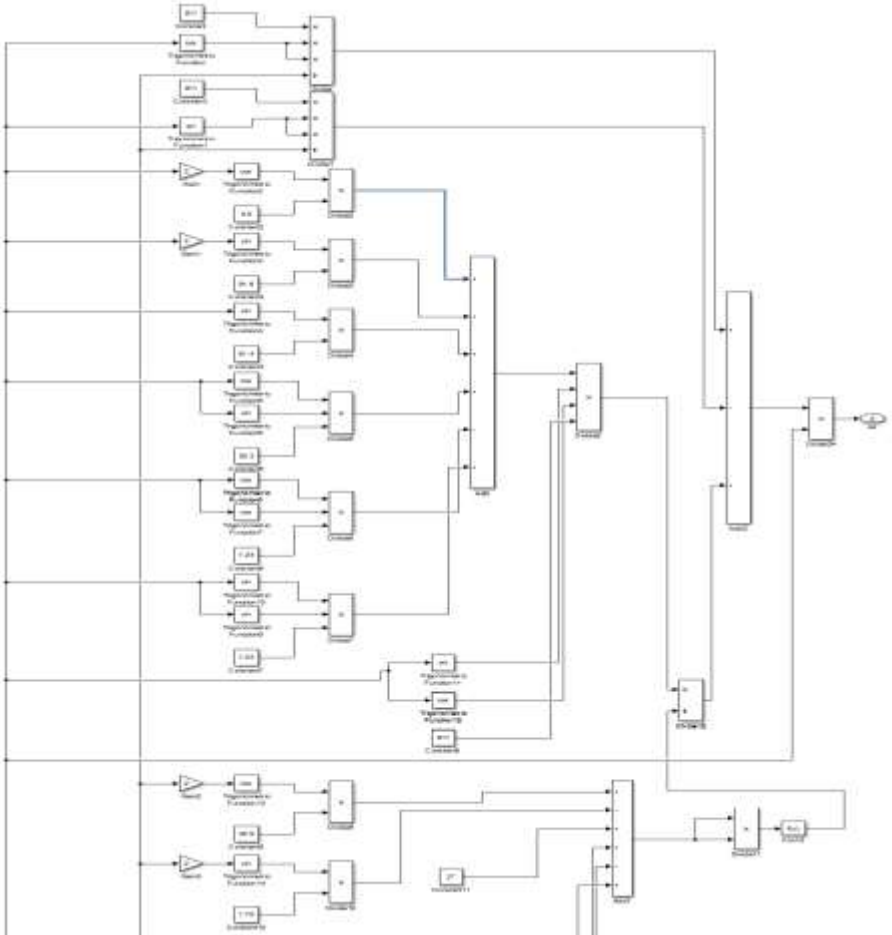


Gambar A 1 Hasil plotting pemodelan kinematis

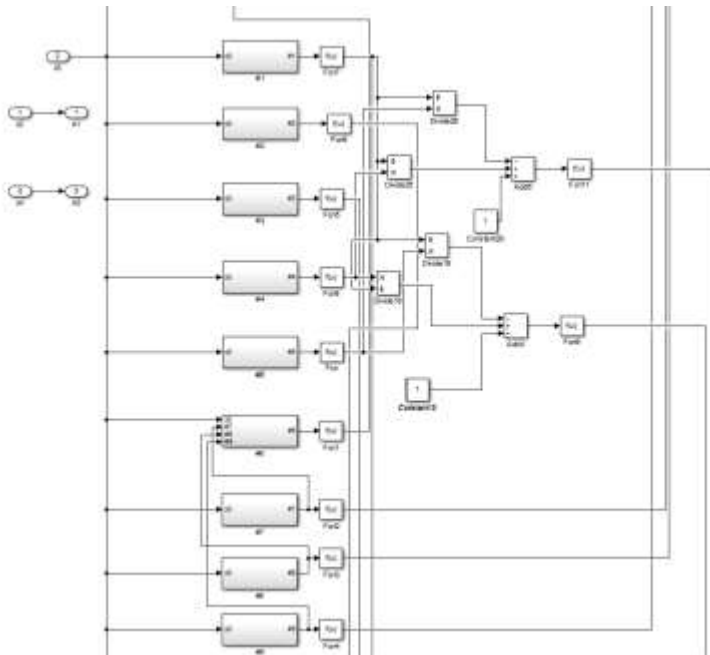
LAMPIRAN B

BLOK DIAGRAM *PLANT*

Pada lampiran ini akan ditampilkan isi dari blok *plant* sebagai model sistem RCWS 12,7mm. Isi dari Blok *plant* mengikuti hasil perhitungan pada matlab. Matriks A dan B *state space* adalah sebagai berikut.



Gambar B 1 Matriks A1



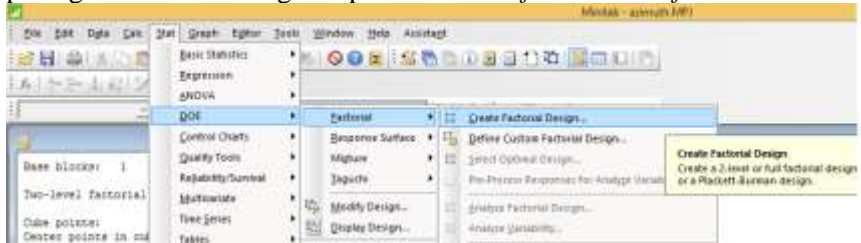
Gambar B 2 Matrik A2

LAMPIRAN C

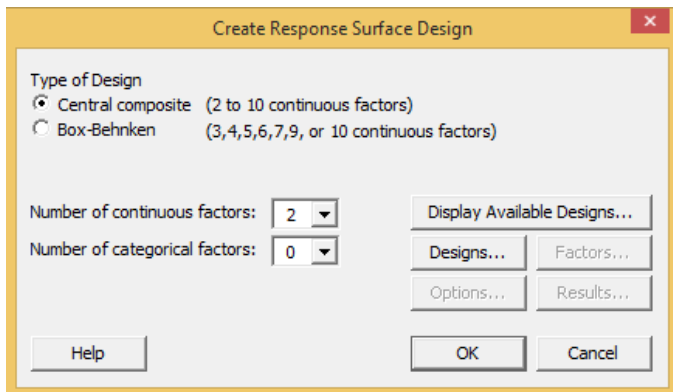
ANALISIS REGRESI dan OPTIMASI

Pada lampiran ini akan ditampilkan cara melakukan regresi dengan menggunakan perangkat lunak Minitab dan optimasi GRG non linear dengan bantuan *solver* pada Microsoft Excel. Hasil dari regresi dan optimasi juga akan ditampilkan.

Dalam melakukan analisis regresi dibutuhkan *design of experiment* (DOE). Pembuatan DOE ini menggunakan bantuan Minitab. Langkah-langkah untuk membuat DOE dapat dilihat pada gambar C1. Isi angka 3 pada *number of continuous factors*.



Gambar C 1 Langkah-langkah membuat DOE



Gambar C 2 Halaman *Create Response Surface Design*.

Pada halaman *Create Response Surface Design*, isikan angka 3 pada kolom *continuous factors*. Pada *Display Available Designs*

pilih *design Central composite full unblocked* dengan *continuous Factors* 3. Buka halaman *design*, pastikan nilai *default Alpha* adalah 1,682 pada nilai blok 1, lalu klik ok. Buka halaman *Factors* lalu ubah nilai maksimum dan minimum untuk masing-masing parameter yaitu nilai Kp, Ki dan Kd. Pendefinisian level pada halaman *Factors* dipilih *axial points*. Pada halaman *Options* lepaskan centang pada *randomize runs*. Apabila semua sudah dilakukan tekan OK. Hasil dari DOE *response surface* untuk gerak Azimuth dapat dilihat pada tabel C1. DOE ini dibuat untuk masing-masing gerak azimuth dan elevasi.

Tabel C 1 DOE *Response Surface*

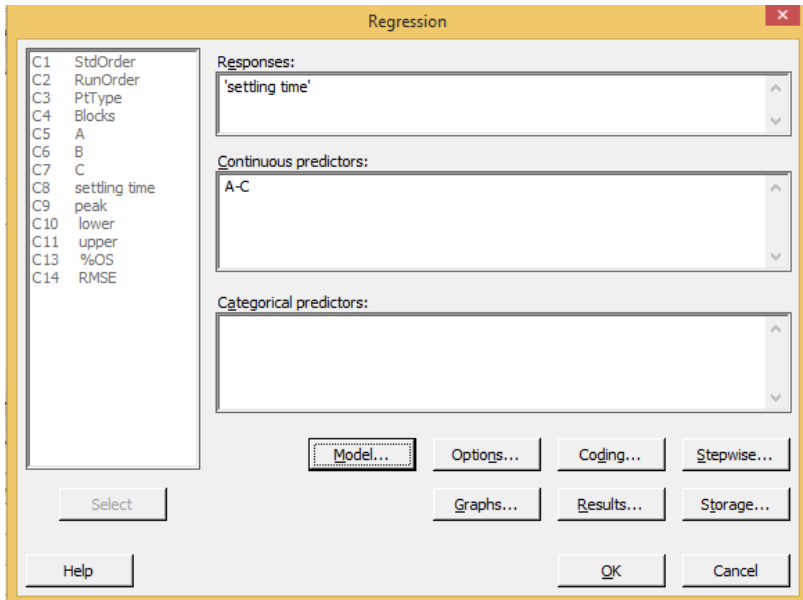
No	Kp	Ki	Kd
1	46.082	24.055	1.811
2	63.918	24.055	1.811
3	46.082	35.945	1.811
4	63.918	35.945	1.811
5	46.082	24.055	4.189
6	63.918	24.055	4.189
7	46.082	35.945	4.189
8	63.918	35.945	4.189
9	40.00177	30	3
10	69.99823	30	3
11	55	20.00174	3
12	55	39.99826	3
13	55	30	1.000348
14	55	30	4.999652
15	55	30	3
16	55	30	3
17	55	30	3
18	55	30	3

19	55	30	3
20	55	30	3

Setelah membuat DOE, lakukan Regresi. Langkah-langkah pada Minitab untuk memunculkan *Regression* dapat dilihat pada gambar C3. Halaman *Regression* dapat dilihat pada gambar C4. Kolom *Responses* diisi dengan respon yang ingin dicari persamaanya, dan kolom *continuous predictors* diisi dengan kolom Kp, Ki dan Kd. Pada halaman *model*, pilih semua *predictors* lalu tambahkan interaksi antar prediktor dan *Terms through order* dengan angka 2. Lalu klik ok. Pada kolom *session* akan tertera persamaan regresi. Salah satu hasil dari analisis regresi dapat dilihat pada gambar C5. Persamaan ini kemudian di salin dan pindahkan pada Microsoft Excel untuk melakukan optimasi.



Gambar C 3 Langkah-langkah memunculkan halaman *Regression*.



Gambar C 4 Halaman Regression

Regression Equation

$$\text{settling time} = 3.74 + 0.1955 A - 0.2773 B - 0.411 C - 0.000249 A^2 + 0.0500 C^2 - 0.002942 A*B - 0.00012 A*C + 0.0005$$

Gambar C 5 Hasil analisis regresi untuk respon *settling time* pada gerak azimuth

Persamaan hasil regresi diletakkan pada kolom tertentu, kolom ini yang kemudian disebut dengan fungsi objektif. Optimasi yang digunakan adalah GRG Nonlinear. Optimasi ini menggunakan fungsi *solver* yang ada pada Microsoft Excel. Untuk mengeluarkannya ada pada tautan *Data* dan pilih *solver*. Halaman *solver* dapat dilihat pada gambar C6 Kolom yang berfungsi sebagai fungsi objektif dimasukkan pada kolom *Set Objective* dan kemudian pilih fungsinya apakah ingin memaksimalkan, meminimalkan atau untuk mencapai nilai tertentu. Parameter yang mempengaruhi fungsi objektif juga didefinisikan pada

kolom *By Changing Variable Cells*. Batasan-batasan baik untuk nilai parameter dan nilai respon ditambahkan pada kolom *constraints*. Pilih metode penyelesaian GRG Nonlinear. Setelah semua terpenuhi maka klik *solve* dan angka optimal untuk masing-masing parameter akan tertera pada sel parameter.

Solver Parameters

Set Objective:

To: ☐ Max ☒ Min ☐ Value Of:

By Changing Variable Cells:

Subject to the Constraints:

-
-
-
-

☒ Make Unconstrained Variables Non-Negative

Select a Solving Method:

Solving Method
 Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.

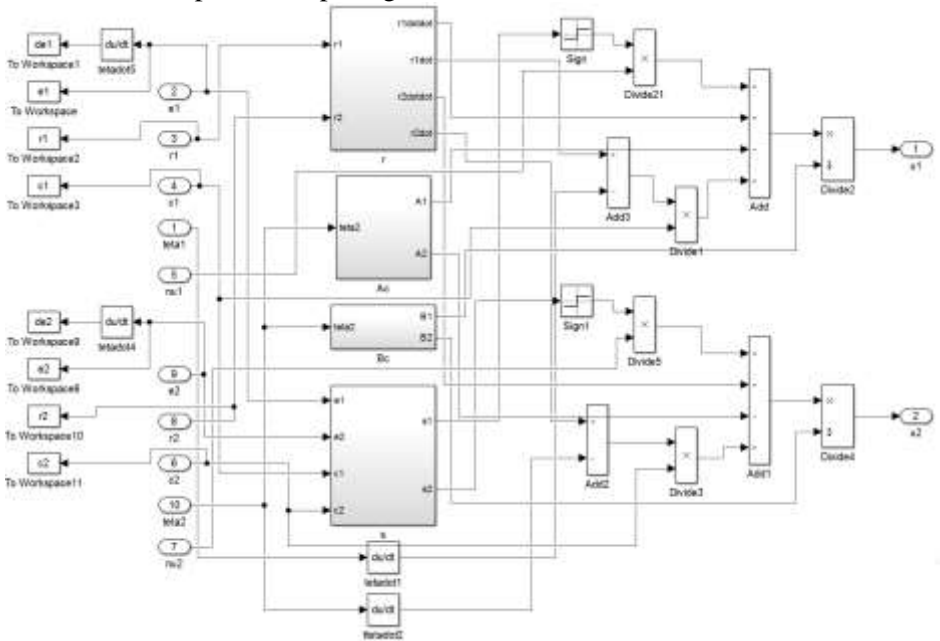
Gambar C 6 Halaman *solver* pada Microsoft Excel

(Halaman ini sengaja dikosongkan.)

LAMPIRAN D

BLOK DIAGRAM KONTROL *SLIDING MODE CONTROL*

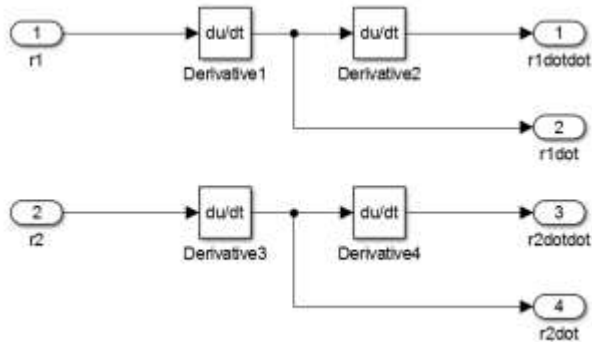
Pada lampiran ini akan ditampilkan isi dari pada blok controller pada kendali *sliding mode control*. Blok diagram dari pengendali SMC dapat dilihat pada gambar 4.4. Isi dari blok control dapat dilihat pada gambar D1.



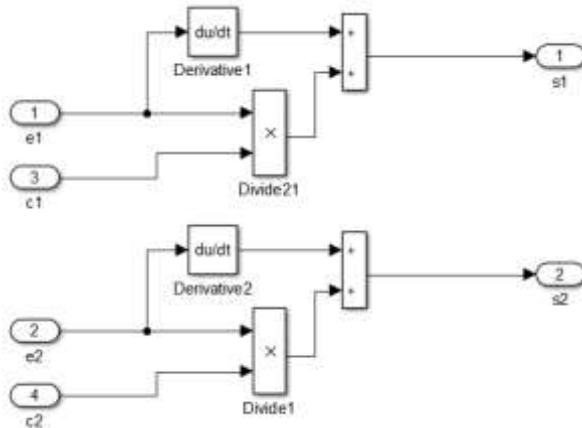
Gambar D 1 Blok diagram kontrol *sliding mode control*.

Isi dari blok kontrol ini membutuhkan 5 input untuk masing-masing gerak sudut yaitu *error* (e), nilai *reference* (r), nilai konstanta λ , besar sudut dan nilai konstanta η . Terdapat 4 subsistem untuk meringkas blok diagram. Subsistem A_c dan B_c diambil dari subsistem *plant*. Subsistem A_c menghilangkan masukan x_2 dan x_4 serta keluaran A_2 dan A_4 , sedangkan untuk

blok Bc juga menghilangkan masukan U1 dan U2. Subsistem r dan s secara berurutan dapat dilihat pada gambar D2 dan D3. Blok *to workspace* digunakan untuk mengeluarkan nilai kedalam Matlab dan nilainya digunakan untuk mengeplot grafik trayektori e dan \dot{e} . Keluaran dari blok kontrol ini adalah plant input.



Gambar D 2 Subsistem r



Gambar D 3 Subsistem s

LAMPIRAN E

CODING PADA MATLAB

Berikut akan ditampilkan koding keseluruhan perhitungan dari penelitian ini.

```
clear;
clc;
syms d1 d2 a1 a2 teta1 alpha1 teta2 alpha2;
%input
%m1 m2 d1 d2 alpha1 alpha2 a1 a2 g inertia
m1 = sym(67.24, 'd');
m2 = sym(9.80, 'd');
d1 = sym(0.389, 'd');
d2 = sym(0, 'd');
a1 = sym(0.0975, 'd');
a2 = sym(1.07578, 'd');
%g berupa vektor
g = sym([0,0,9.8], 'd');
alpha1 = sym(deg2rad(270), 'd');
alpha2 = sym(deg2rad(0), 'd');

%inertia diambil dari "inertia at center of
mass"
i1 = sym([1 -0.02 -0.01; -0.022 -0.63 0.78; 0.01
0.78 -0.63], 'd');
i2 = sym([0 0 1; -0.07 -1 0; 1 -0.07 0], 'd');

%transformation matrix
T11 = [1 0 0 0; 0 1 0 0; 0 0 1 d1; 0 0 0 1];
T12 = [cos(teta1) -sin(teta1) 0 0; sin(teta1)
cos(teta1) 0 0; 0 0 1 0; 0 0 0 1];
T13 = [1 0 0 a1; 0 1 0 0; 0 0 1 0; 0 0 0 1];
T14 = [1 0 0 0 ; 0 cos(alpha1) -sin(alpha1) 0; 0
sin(alpha1) cos(alpha1) 0; 0 0 0 1];

T21 = [1 0 0 0; 0 1 0 0; 0 0 1 d2; 0 0 0 1];
T22 = [cos(teta2) -sin(teta2) 0 0; sin(teta2)
cos(teta2) 0 0; 0 0 1 0; 0 0 0 1];
T23 = [1 0 0 a2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
```

```
T24 = [1 0 0 0 ; 0 cos(alpha2) -sin(alpha2) 0; 0
sin(alpha2) cos(alpha2) 0; 0 0 0 1];
```

```
A1 = T11*T12*T13*T14;
```

```
A2 = T21*T22*T23*T24;
```

```
TDH = A1*A2;
```

```
syms teta psi px py pz;
```

```
T31 = [cos(teta) -sin(teta) 0 0; sin(teta)
cos(teta) 0 0; 0 0 1 0; 0 0 0 1];
```

```
T32 = [cos(psi) 0 -sin(psi) 0 ; 0 1 0 0;
sin(psi) 0 cos(psi) 0; 0 0 0 1];
```

```
Arot = T31*T32;
```

```
px = (a1+(a2*cos(psi)))*cos(teta);
```

```
py = (a1+(a2*cos(psi)))*sin(teta);
```

```
pz = d1 +a2*sin(teta);
```

```
Apos = [1 0 0 px; 0 1 0 py; 0 0 1 pz; 0 0 0 0];
```

```
TEE = Apos *Arot;
```

```
sol2 = solve(sin(alpha1)*sin(teta2) == sin(psi),
teta2);
```

```
sol1 = solve(cos(tetal)*cos(teta2) -
cos(alpha1)*sin(tetal)*sin(teta2) ==
cos(psi)*cos(teta), tetal) ;
```

```
%menentukan matrix Rotasi
```

```
R01= A1(1:3,1:3);
```

```
R12= A2(1:3,1:3);
```

```
R02= TDH(1:3,1:3);
```

```
%menentukan matrix Z
```

```
Z0= [0;0;1];
```

```
Z1= R01*[0;0;1];
```

```
%menentukan matrix Posisi
```

```
P01= ([1]*[a1*cos(tetal); a1*sin(tetal); 0])/2;
```



```

P12= (R01*[a2*cos(teta2); a2*sin(teta2); 0])/2;
P02= ([1]*[a1*cos(tetal); a1*sin(tetal); 0])+
P12;

%inertia link%
%transpose matrix%
R01trans = transpose(R01);
R02trans = transpose(R02);

%hasil inertia
I1 = R01*i1*R01trans;
I2 = R02*i2*R02trans;

%jacobian (Jvij)
Jv11 = cross(Z0,P01);
Jv1 = [Jv11 zeros(size(Jv11,1),1)];
Jv21 = cross(Z0,P02);
Jv22 = cross(Z1,P12);
Jv2 = [Jv21, Jv22];
Jw1 = [0 0;0 0;1 0];
Jw2 = [Z0, Z1];

%Manipulator inertia matrix
M =(transpose(Jv1)*m1*Jv1 +
transpose(Jw1)*I1*Jw1 + transpose(Jv2)*m2*Jv2 +
transpose(Jw2)*I2*Jw2);

%Velocity coupling vector
syms tetadot1 tetadot2
tetadot = [tetadot1;tetadot2];
teta = [tetal;teta2];

for j = 1:1:2;
    for k = 1:1:2;
        hasilv1(j,k) = (diff(M(1,j),teta(k))-
(0.5.*(diff(M(j,k),teta(1))))).*tetadot(j).*teta
dot(k);
    end
end

```

```

V1 = hasilv1(1,1) + hasilv1(1,2) + hasilv1(2,1)
+ hasilv1(2,2);

for j = 1:1:2;
    for k = 1:1:2;
        hasilv2(j,k) = (diff(M(2,j),teta(k))-
(0.5.*(diff(M(j,k),teta(2))))).*tetadot(j).*teta
dot(k);
    end
end
V2 = hasilv2(1,1) + hasilv2(1,2) + hasilv2(2,1)
+ hasilv2(2,2);

%Gravitational vector
syms Jv m
Jv = [Jv1, Jv2];
m = [m1;m2];

G1 = ((m1*g*Jv11)+(m2*g*Jv21));
G2 = m2*g*Jv22;

%Lagrange's equation of motion
%assumption: no external force and joint
friction is negligible

syms tetadotdot tetadotdot1 tetadotdot2
tetadotdot=[tetadotdot1;tetadotdot2];

for j=1:1:2
    sblmtorsil(j)=M(1,j)*tetadotdot(j);
end
sigmasblmtorsil = sblmtorsil(1)+sblmtorsil(2);
torsilmatrix = sigmasblmtorsil + V1 +G1;

for j=1:1:2
    sblmtorsi2(j)=M(2,j)*tetadotdot(j);
end
sigmasblmtorsi2 = sblmtorsi2(1)+sblmtorsi2(2);
torsii2matrix = sigmasblmtorsi2 + V2 +G2;

```

```

%memanggil persamaan torsi
torsil = torsilmatrix;
torsil2 = torsil2matrix;

%penyederhanaan
torsilsimp= simplify(torsil);
torsil2simp= simplify(torsil2);

cutoffup = 10^-3;
cutofflw = -10^-3;
[C1, T1] = coeffs(torsilsimp);
[C2, T2] = coeffs(torsil2simp);
C1(cutofflw<C1 & C1<cutoffup) = 0;
C2(cutofflw<C2 & C2<cutoffup) = 0;
torsilfixed = sum(C1.*T1);
torsil2fixed = sum(C2.*T2);

%pembulatan
torsilbul= vpa(torsilfixed,3);
torsil2bul= vpa(torsil2fixed,3);

%koefftomatrix
%kdot21 artinya koeff dan variabel untuk
tetadot2 di torsi 1
[kdotdot1,vdotdot1] =
coeffs(torsilbul,tetadotdot1);
[kdotdot2,vdotdot2] =
coeffs(torsil2bul,tetadotdot2);

%persamaan teta
syms torsilinput torsil2input %ptetadotdot1c
ptetadotdot1= vpa((-
kdotdot1(1,2)+torsilinput)/kdotdot1(1,1),3);
ptetadotdot2= vpa((-
kdotdot2(1,2)+torsil2input)/kdotdot2(1,1),3);
ptetadotdot1A=vpa((-
kdotdot1(1,2))/kdotdot1(1,1),3);

```

```

ptetadotdot2A=vpa((-
kdotdot2(1,2))/kdotdot2(1,1),3);

%substitusi tetadotdot coba
pstetadotdot1 =
subs(ptetadotdot1,sym('tetadotdot2'),ptetadotdot
2);
pstetadotdot2 =
subs(ptetadotdot2,sym('tetadotdot1'),ptetadotdot
1);
pstetadotdot1A =
subs(ptetadotdot1A,sym('tetadotdot2'),ptetadotdo
t2A);
pstetadotdot2A =
subs(ptetadotdot2A,sym('tetadotdot1'),ptetadotdo
t1A);

pertetadotdot1 = tetadotdot1 == pstetadotdot1;
pertetadotdot2 = tetadotdot2 == pstetadotdot2;
pertetadotdot1A = tetadotdot1 == pstetadotdot1A;
pertetadotdot2A = tetadotdot2 == pstetadotdot2A;

%solv persamaan teta
psstetadotdot1 = solve(pertetadotdot1,
tetadotdot1);
psstetadotdot2 = solve(pertetadotdot2,
tetadotdot2);
% pssstetadotdot2 =
psstetadotdot1A = solve(pertetadotdot1A,
tetadotdot1);
psstetadotdot2A = solve(pertetadotdot2A,
tetadotdot2);

%ganti nama variable
syms x1 x2 x3 x4 dx1 dx2 dx3 dx4 u1 u2
u = [u1;u2];
y = [x1,x3];
pdx2s = vpa(subs(psstetadotdot1,
[sym('teta1'),sym('tetadot1'),sym('tetadotdot1')

```

```

, sym('teta2'), sym('tetadot2'), sym('tetadotdot2')
, sym('torsilinput'), sym('torsilinput')], [sym('x1
'), sym('x2'), sym('dx2'), sym('x3'), sym('x4'), sym(
'dx4'), sym('u1'), sym('u2')]), 3);
pdx4s = vpa(subs(psstetadotdot2,
[sym('teta1'), sym('tetadot1'), sym('tetadotdot1')
, sym('teta2'), sym('tetadot2'), sym('tetadotdot2')
, sym('torsilinput'), sym('torsilinput')], [sym('x1
'), sym('x2'), sym('dx2'), sym('x3'), sym('x4'), sym(
'dx4'), sym('u2'), sym('u1')]), 3);
pdx2As = vpa(subs(psstetadotdot1A,
[sym('teta1'), sym('tetadot1'), sym('tetadotdot1')
, sym('teta2'), sym('tetadot2'), sym('tetadotdot2')
, sym('torsilinput'), sym('torsilinput')], [sym('x1
'), sym('x2'), sym('dx2'), sym('x3'), sym('x4'), sym(
'dx4'), sym('u1'), sym('u2')]), 3);
pdx4As = vpa(subs(psstetadotdot2A,
[sym('teta1'), sym('tetadot1'), sym('tetadotdot1')
, sym('teta2'), sym('tetadot2'), sym('tetadotdot2')
, sym('torsilinput'), sym('torsilinput')], [sym('x1
'), sym('x2'), sym('dx2'), sym('x3'), sym('x4'), sym(
'dx4'), sym('u2'), sym('u1')]), 3);

%nominal points untuk x2 x4
pdx2 = vpa(subs(pdx2s,
[sym('x2'), sym('x4')], [0,0]), 3);
pdx4 = vpa(subs(pdx4s,
[sym('x2'), sym('x4')], [0,0]), 3);
pdx2A = vpa(subs(pdx2As,
[sym('x2'), sym('x4')], [0,0]), 3);
pdx4A = vpa(subs(pdx4As,
[sym('x2'), sym('x4')], [0,0]), 3);

%fungsi non-linear
dx1 = x2;
dx2 = pdx2;
dx3= x4;
dx4 = pdx4;
%fungsi non-linear A

```

```

dx1A = x2;
dx2A = pdx2A;
dx3A= x4;
dx4A = pdx4A;

%jacobian
aj=
vpa(jacobian([dx1,dx2A,dx3,dx4A],[x1,x2,x3,x4]),
3);
bj= vpa(jacobian([dx1,dx2,dx3,dx4],[u1,u2]),3);
cj= vpa(jacobian(y,[x1,x2,x3,x4]),3);
dj= vpa(jacobian(y,[u1,u2]),3);

```

LAMPIRAN F

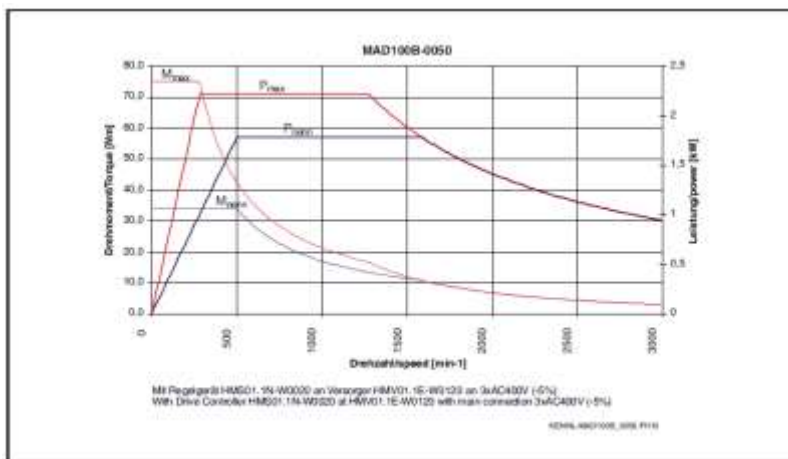
TECHNICAL DATA SHEET MOTOR LISTRIK

Pada lampiran ini akan ditampilkan *technical data sheet* untuk motor listrik. Motor listrik yang digunakan adalah Rexroth IndraDyn A dengan tipe MAD100B dan jumlah lilitan sebanyak 50 lilitan. Lembar data motor dengan tipe MAD100B dapat dilihat pada gambar F1. Kurva karakteristik motor MAD100B dengan 50 lilitan juga dapat dilihat pada gambar F2.

Description	Symbol	Unit	MAD100B					
Motor data ¹⁾								
Winding			3040	0100	0150	0200	0250	
Rated torque	M_{rated}	Nm	34	31	29	26	25	
Rated speed	n_{rated}	rpm	500	1000	1500	2000	2500	
Rated power	P_{rated}	W	1.8	3.2	4.4	5.9	6.5	
Rated current	I_{rated}	A	5.3	8.9	11.8	14.6	16.2	
Continuous torque at standard	M_{L}	Nm	—	34	31	26	25	
Starting speed	n_s	rpm	—	950	1000	1000	2000	
Continuous current at standard	I_{L}	A	—	9.4	12.2	15.2	16.8	
Maximum speed	M_{max}	Nm	75	75	65	60	62	
Torque constant at 23°C	$K_{\text{M, rated}}$	Nm/A	7.66	4.31	3.81	2.41	2.11	
Number of pole pairs	p		3	—	—	—	—	
Min. cross section of power cable ²⁾	A_{min}	mm²	—	—	—	2.5	—	
Rotor moment of inertia ³⁾	J_{rot}	kgm²	—	—	—	0.019	—	
Mass of motor ⁴⁾	m	kg	—	—	—	3.8	—	
Maximum speed with — Standard — crimping attachment — standard — High-speed	P_{max}	rpm	3000 3000 3000	6000 6000 6000	— 6000 6000	— — 8000	8000 8000 10000 ⁵⁾	
Thermal time constant	T_{th}	min	—	—	—	30	—	
Duty cycle time (50–40%)	T_{D}	min	—	—	—	14	—	
Noise level ⁵⁾	P_{noise}	dB(A)	—	—	—	70 (+3)	—	
Permissible ambient temperature	T_{amb}	°C	—	—	—	0...+40	—	
Permissible storage and transport temperature	T_{store}	°C	—	—	—	-20...+60	—	
Insulation class according to DIN VDE 0530-1			—	—	—	F	—	
International Protection class			—	—	—	IP55	—	
Holding brake (optional)								
Transmittable torque	M_{H}	Nm	30	—	—	24	—	
Connection voltage	U_{H}	V	—	—	—	DC 24 ± 10 %	—	
Rated current	I_{H}	A	0.9	—	—	1.1	—	
Moment of inertia	J_{H}	kgm²	—	—	—	0.00386	—	
Max. permissible braking energy	W_{max}	Ws	—	—	—	20000	—	
Disengagement time	t_{D}	ms	60	—	—	60	—	
Engagement time	t_{E}	ms	42	—	—	30	—	
Maximum speed of brake	n_{max}	rpm	10000	—	—	10000	—	
Mass of brake	m	kg	2	—	—	1.6	—	
Blower								
Air current			—	—	—	B → A, blowing	—	
Connection voltage	U_{B}	V	—	—	—	3 × 400V ± 10 %, 50/60 Hz	—	
Power consumption	P_{B}	VA	—	—	—	83...160	—	
Blower current ⁶⁾	I_{B}	A	—	—	—	0.12	—	
Average air flow rate	\dot{V}	m³/h	—	—	—	270	—	

¹⁾ Values determined according to IEC 60034-1. Current and voltage specified as root-mean-square values.
²⁾ Rated for cable assemblies with current carrying capacity according to VDE0298-1 (1992) and installation type B2 according to EN 60204-1 (1993) at 40°C ambient temperature.
³⁾ Values without holding brake.
⁴⁾ Values without holding brake, with blower.
⁵⁾ at 1m distance, with PWM = 4 kHz.
⁶⁾ From I_{B} + 20%, blowers should be monitored.
⁷⁾ Value is without a holding brake. This value may be limited by a holding brake.

Gambar F 1 Lembar data MAD100B



Gambar F 2 Kurva karakteristik MAD100B dengan 50 lilitan.

BIODATA PENULIS



Penulis yang memiliki nama lengkap Hans Kristian lahir pada 26 Mei 1995 di Surabaya. Penulis merupakan anak ke 2 dari 2 bersaudara. Penulis menempuh pendidikan formal hingga jenjang S1 di Surabaya, yaitu TKK Santa Clara, SDK Santa Clara, SMPK Santa Clara, SMAK St Louis 1 dan melanjutkan jenjang S1 di Institut Teknologi Sepuluh Nopember (ITS) jurusan Teknik Mesin dengan nomor induk 2113100005.

Selama masa studi di ITS, penulis aktif dalam kegiatan Paduan suara mahasiswa ITS dan merupakan anggota aktif serta menjadi salah satu staf Pengembangan Sumber Daya Mahasiswa (PSDM) periode 2014/2015 dan 2015/2016. Dalam studinya, penulis menjadi anggota lab otomasi dan menyelesaikan penulisan tugas akhir dibawah bimbingan 2 dosen yaitu Ir. Bambang Pramujati, MSc.Eng, PhD. dan Hendro Nurhadi, Dipl. Ing., Ph.D. Penulis dapat dihubungi melalui alamat email sebagai berikut: hanskristiantj@gmail.com.